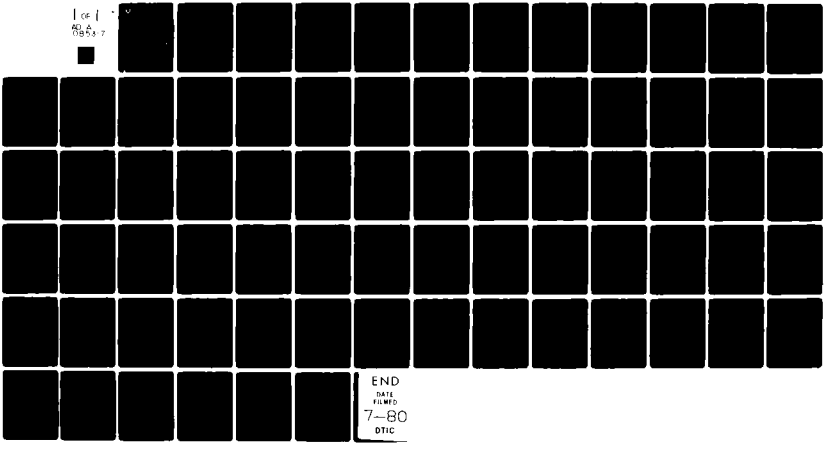
AD-A085 317 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND FO--ETC F/G 9/2
DOCUMENTATION OF SOFTWARE IN THE OL-192 PIBAL PROGRAM.(U)
APR 80 R BELLUCCI

UNCLASSIFIED DELCS-TR-80-1 NL



END
DATE FILMED
7-80
DTIC

LEVEL II

(12)

RESEARCH AND DEVELOPMENT TECHNICAL REPORT

DELCS-TR-80-I

DOCUMENTATION OF SOFTWARE IN THE OL-192 PIBAL PROGRAM

Raymond Bellucci
COMBAT SURVEILLANCE & TARGET ACQUISITION LABORATORY

April 1980

DISTRIBUTION STATEMENT

Approved for public release;
distribution unlimited.

ERADCOM

US ARMY ELECTRONICS RESEARCH & DEVELOPMENT COMMAND
FORT MONMOUTH, NEW JERSEY 07703

NOTICES

Disclaimers

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: DELCS-TR-80-1	2. GOVT ACCESSION NO.: AD-A085 317
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): DOCUMENTATION OF SOFTWARE IN THE OL-192 PIBAL PROGRAM.	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): Raymond Bellucci	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Combat Surveillance & Target Acquisition Laboratory, ATTN: DELCS-S, Fort Monmouth, NJ 07703	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 674726.12.511.04.11
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Electronics Research and Development Command, ATTN: DRDEL-PAO, Adelphi, MD 20783	12. REPORT DATE: April 1980
	13. NUMBER OF PAGES: 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Program
Meteorological Data Reduction
Ballistic Meteorological Message
Computer Meteorological Message
Sound Ranging Message

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the software contained in the OL-192 PIBAL Program, and is intended to supply complete program description, file definitions, flow charts, and program listing. The program was written for the Hewlett Packard 9825A Calculator.

The OL-192 PIBAL Program is used whenever the electronic capability of the Met station is inoperative either through equipment failure or limited

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

1521

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract - continued
frequency allocations. It quickly converts time and angular data into real time meteorological messages for the region of interest.

Tables of unweighted temperatures and pressures for Central Europe, Scandinavia, Italy and Spain, Greece and E. Mediterranean, North Africa, South Africa, Western Russia, Siberian, Korea, SE Asia and the tropics, and Alaska and Polar regions are included in separate program tapes.

The tables are entered using the true surface density as percent of standard.

CONTENTS

FIGURES

CONTENTS

Page

TABLES

SUMMARY

The purpose of this task was to provide the artillery with a program for computing the meteorological messages from pilot balloon observations. A pilot balloon is a small balloon whose ascent is followed by a theodolite to obtain data for computation of the speed and direction of the winds in the upper air.

This document is a description of the software contained in the OL-192 PIBAL Meteorological Data Reduction Program and is intended to supply complete program description, file definitions, variable descriptions, flow charts, and program listings. This program was written for the Hewlett Packard 9825A Calculator, and is the basis for the project which is now under configuration management. Any changes to this program must be accomplished by an Engineering Change Proposal.

Program Identification Data

Product Improvement Plan: 78-07-020-I

Program Title: PIBAL Meteorological Data Reduction Program for the Hewlett Packard 9825A Calculator.

Programming Personnel: Raymond Bellucci

Originating Department: Combat Surveillance & Target Acquisition Laboratory, ERADCOM, Fort Monmouth, New Jersey 07703

1. PROGRAM DESCRIPTION

Purpose: A PIBAL solution for determining meteorological messages from single theodolite measurements has been added to the software of the original OL-192 program. PIBAL measurements are required by the Artillery for developing Sound Ranging messages and for use in areas where it is impossible to use electronic equipment.

Use: The PIBAL version of the OL-192 is used by the Artillery meteorological section to convert the angular data obtained by tracking a pilot balloon by a single theodolite ML-474. Input is entered from the calculator's console for near real time computation. Departure tables for each of 11 regions of the world are stored in the program cassettes, providing the temperature-pressure-density information for the meteorological messages.

Physical Description

Hewlett-Packard 9825A Calculator

Displays:

Light Emitting Diodes (LED)	- 32-character data display to prompt the operator.
Strip Printer	- Prints 16-character wide hard copy listing of programs, entries and messages.
Live Keyboard	- Permits use of the computer while a program is running.
Audio Beeper	- Alerts operator for data inputs.

Remex Reader/Perforator

Tape Reader:

Tape Material	:	8-channel, 1-inch wide ASCII, Mylar or paper.
Speed	:	300 characters per second.
Tape Form	:	Loop or strip.
Tape Direction	:	Left to right.

Perforator:

Tape Material	:	8-channel, 1-inch wide, ASCII, Mylar or paper. 5-channel, 11/16-inch wide, BAUDOT, Mylar or paper.
Speed	:	120 characters per second.
Tape Form	:	1,200-foot roll (2-inch core).

Reader/Perforator Rewind
Winds punch data tapes for ease of storage.

LED Display and Audible Beep Interpretation.

Types of displays and beeps

1. Verification of data entered prompt: displays data for verification and ends with a question mark (?). The YES and NO keys are activated and there is one audible beep.

2. Entry of data prompt: displays ENTER, type of data entry, and a range of entry data values. There are two audible beeps.

3. Action prompt: displays the type of action required. There are three audible beeps.

Flag Definitions (True/False):

Flag 0: Optional output.

Flag 1: Figures output on BAUDOT code (TTY-76)/no figures output.

Flag 2: Letters output on BAUDOT code (TTY-76)/no letters output.

Flag 3: No delete/delete.

Flag 4: Paper tape reader-perforator on/off.

Flag 5: Figures mode on BAUDOT code (TTY-76)/no figures mode.

Flag 6: Letters mode on BAUDOT code (TTY-76)/no letters mode.

Flag 7: TTY-76 code (BAUDOT).

Flag 8: UGC-74 code (ASCII).

Flag 9: Continue flight/flight.

a. System Flow

The general flow of the PIBAL program is controlled by a Master Supervisory File and the Special Function Keys. When the calculator is turned on and the program cartridge is seated, the calculator will automatically load an initiating program and begin running the program. The program prints the program title, version date, tape number and counter number. It then loads the functional key definitions from file 1 (see Table 1 for description of the special function keys). The initiating program prompts the operator to press one of the upper row keys that activates the PIBAL routine. The first 6 functional keys are:

1. f0 START - PRESS PIBAL.
2. f1 CONT - PRESS PIBAL.
3. f2 LAST FLT - PRESS PIBAL.

4.	f3	P. TAPE	- PRESS PIBAL.
5.	f4	TAC FIRE	- PRESS PIBAL.
6.	f5	PIBAL	- PIBAL PROGRAM LOADED.

The other six functional keys are activated by the individual routines. They are:

7.	f6	INFO	- PRESS PIBAL.
8.	f7	OUTPUT	- LAST FLIGHT ENTERED.
9.	f8	DELETE	- Delete message.
10.	f9	NO	- No answer key.
11.	f10	YES	- Yes answer key.
12.	f11	DATA IN	- Used to input data into calculator.

An operator's flow chart and a system flow chart showing the interconnection of the routines is in section 4.a (pg. 23).

The OL-192 PIBAL program is divided into three logical sections.

1. Preflight (surface data).

2. Flight (inflight data reduction).

3. Output (meteorological messages).

The preflight section is entered when the operator presses the "PIBAL" Key. After the preflight section, the program enters the flight section. The Output section is entered automatically after the operator keys in an * during the flight section or when 20 levels of data have been entered. Pressing the OUTPUT key any time prior to entering an * results in a last flight OUTPUT.

b. Preflight Section

The preflight section is entered by pressing the "PIBAL" key. Pressing the "START" or any of the upper row keys prompts the operator to press the "PIBAL" key. The preflight routine prompts the operator through entry of the Date-Time Group, surveys of the launch site, surface data, balloon weight and gas used for balloon inflation.

(1) Date-Time Group

The Date-Time Group routine prompts the operator to enter the year, month, day and time in GMT (ZULU) and the Local Standard Time and assembles the Date-Time Group. The group is used to update the flight date, survey date and last flight date. The DATE-TIME Group is stored in D$.

(2) Survey Data

In this routine the operator enters the station elevation, latitude, longitude and quadrant (see Table 2). The routine prints the survey data for verification and flight record. It updates P$ and the flight launch site date 0$. In the messages, the headers contain the site identification.

(3) Launch Data

This routine calculates the geopotential height constant, enters azimuth offset, horizontal distance offset, surface wind direction, surface wind speed, surface pressure and surface wet and dry bulb temperatures. The balloon weight and the type of gas used for inflating the balloon is also entered in this routine. These values are verified and stored in array A(*). In the flight section they will be stored in array F(*).

c. Flight Section

All the flight data are entered in this section. These data include time, elevation angle, and azimuth angle of the balloon in flight. The program is initiated after the balloon weight and type of gas used for inflation of the balloon are entered. If the program determines that a zone height has been passed, it calculates the zone level times, azimuth and elevation angles from an interpolation routine which stores these values in array H(*).

(1) Zone Information Section

This section is entered into automatically after 20 levels of data have been inputted or if an * is entered during the DATA-IN procedure. Zone level data consisting of wind speed and direction for each zone centered at the midpoint of the zone together with the time, elevation and azimuth angles of the balloon at the top of the zone are printed out. This information provides the operator with a check on the validity of his data.

(2) Departure Tables Section

At this time the operator selects the region of interest by entering the region number. The departure table for the selected region is printed out on hard copy.

(3) Output Type

At this time, the operator decides whether he wants a punched teletape copy of the messages. It will be in the TTY-76 (BAUDOT code - 5 channel - 11/16-inch tape: see Table 3). When punching BAUDOT tape, the OUTPUT section converts the ASCII code, used internally by the calculator, to BAUDOT using a string array of 64 characters. The character is looked up in the array H$. The position converts into a number between 0 and 31. The figures and letters mode is determined by the 32-character subarray in which the character is found. If the character is not in the respective mode, the mode key (figures or letters) is punched before punching the character.

d. OUTPUT Section

This section is entered from the Departure Tables section. The Output program produces meteorological messages from the flight data in the flight level array F(*) and the zone level array H(*) tables that were entered in the Flight section. The OUTPUT section computes and outputs the following messages from the current flight:

SOUND RANGING
COMPUTER MET (FADAC M-18)
FALLOUT
BALLISTIC 3
BALLISTIC 2

(1) Sound Ranging Message

The general requirement is to determine the windspeed and direction in four layers from the surface to 800 meters and to determine the virtual temperature at a height of 200 meters. The data transmitted are the effective temperature, effective wind direction, effective windspeed and the time of release. The effective temperature is the sonic temperature at 200 meters. For the effective wind direction and speed, the sound ranging layer wind calculations are the weighted and summed X, Y wind components for the surface, 200, 400, 600 and 800 meter levels:

(a) The Weighting Factors are:

Structure	Height Surface	200	400	600	800
Normal	0.2	0.5	0.15	0.075	0.075
2	0.4	0	0.3	0.15	0.15
3	0	1.0	0	0	0
4	0	0	1.0	0	0

Normal Structure - The 400-meter layer wind is one to two times the 200-meter layer wind.

Structure 2 - The 400-meter layer wind is greater than two times the 200-meter layer wind.

Structure 3 - The 400-meter layer wind is less than the 200-meter layer wind and within two knots of the surface wind.

Structure 4 - The 400-meter layer wind is less than the 200-meter layer wind and not within two knots of surface wind.

(b) Sound Ranging Message Format

METSRQXXXXXX
YYGGGV±TTTDDDFF
9

METSR Identifying prefix for sound ranging message.

Q Octant of the globe (0 to 8 not 4), 9 if location is coded.

XXXXXX Location of reporting station. The first three digits encode latitude and the last three encode longitude in degrees and tenths of degrees.

YY — Day of month (01 to 31).

GGG — Hour of valid period (GMT) in tens, units and tenths of hours (000 to 239).

V — Length of valid period in hours (1 to 8), 9 equals 12 hours. (Set to Ø).

±TTT — Effective temperature to nearest 1/10th degree C (-499 to +499).

DDD — Effective wind direction in tens of mils (001 to 640). (000 for calm wind).

FF — Effective wind speed in knots (00 to 99).

9 — Message terminator. (Punch output).

(2) Computer Meteorological Message (FADAC M-18)

The computer message differs from the ballistic (NATO) message in that the zoning structure is different, the zone values are not weighted, and the weather elements are reported as true values instead of weighted percent of standard. The elements reported for each of the 26 zones (see Table 5) are wind direction, wind speed, virtual temperature, and the zone midpoint pressure.

Computer Meteorological Message Format

```
METGMQXXXXXX
YYGGGVHHHPPP
ZZDDDFFFTTTTRRRR
xxxxxxxxxxxxxxxx
26DDDFFFTTTTRRRR
9
```

METCM — Identifying prefix for computer message.

Q — Octant of the globe.

XXXXXX — Location of reporting station. The first three digits encode latitude and the last three encode longitude in degrees and tenths of degrees.

YY — Day of month (01 to 31).

GGG — Hour of valid period (GMT) in tens, units and tenths (000 to 239).

V — Length of valid period hours (1 to 8), 9 equals 12 hours. (Set to Ø).

HHH — Altitude of meteorological datum plane in tens of meters above mean sea level.

PPP	Station pressure to nearest millibar (omit thousands digit).
ZZ	Line number for message (00 to 26).
DDD	Zone wind direction in tens of mils (001 to 640) (000 for calm wind).
FFF	Zone wind speed in knots (000 to 999).
TTTT	Zone virtual temperature to nearest 0.1 degree K (0000 to 5000).
RRRR	Zone midpoint pressure in millibars (0000 to 1100).
9	Message terminator (Punch output).
/	Missing data indicator.

(3) FALLOUT Message

The Meteorological Message for fallout contains the average vector wind for each 2,000 meter zone from the surface to a height of 30,000 meters (see Table 5). The average wind for each 2,000-meter zone is reported to the nearest one knot and to the nearest 10 mils. The distance traveled in each fallout zone is calculated from the zone level data and an average speed and direction are computed. Fallout winds are not weighted:

Fallout Message Format

METFMQXXXXXX
YYGGGVHHHTRO
ZZDDDFFF
xxxxxxxx
15DDDFFF
9

METFM	Identifying prefix for fallout message.
Q	Octant of the globe (0 to 8 not 4), 9 if location is coded.
XXXXXX	Location of reporting station. The first three digits encode latitude and the last three encode longitude in degrees and tenths of degrees.
YY	Day of month (01 to 31).
GGG	Hour of valid period (GMT) tens, units and tenths (000 to 239).
V	Length of valid period (1 to 8), 9 equals 12 hours (Set to Ø).

HHH — Altitude of meteorological datum plane in tens of meters above mean sea level (000 to 999).

ZZ — Line number for fallout data (00 to 15).

DDD — Wind direction to nearest 10 mils (001 to 640) (000 for calm winds).

FFF — Wind speed in knots (000 to 999).

9 — Message terminator (Punch output).

/ — Missing data indicator.

TRO — Filler for fallout message.

(4) Ballistic Meteorological Messages

The ballistic message is a measure of the parameters of the atmosphere, a comparison of the current conditions with standard conditions, and a report of the variations in terms of weighted percents of standard (see Tables 6 and 7). The average wind speed and direction for each of the atmospheric zones are determined. The zone values of density and temperature are compared with the standard zone values and variations from the standard are determined. The variations from standard are then weighted according to specified zone weighting factors. These mean weighted quantities are the ballistic values.

Two general categories of trajectories have been established:

Type 3 - surface to surface (see Table 8 for weights).

Type 2 - surface to air (see Table 9 for weights).

Ballistic Message Format:

METBKQXXXXXX
YYGGGVHHHPPP
ZZDDFFTTTRRR
xxxxxxxxxxxx
15DDFFTTTRRR
9

METB — Identifying prefix for ballistic messages.

K — Type of message (the value of K may be 2 for ballistic met 2, or 3 for ballistic met 3).

Q — Octant of the globe (0 to 8 not 4), 9 when station is coded.

XXXXXX — Location of reporting station.

YY — Day of the month (01 to 31).

GGG	Hour of beginning valid period (GMT) tens, units, and tenths (000 to 239).
V	Length of valid period hours (1 to 8), 9 equals 12 hours (Set to Ø).
HHH	Altitude of meteorological datum plane in tens of meters above mean sea level.
PPP	Station pressure in percent of ICAO standard to nearest 0.1 percent (000 to 999).
ZZ	Line number for ballistic information (00 to 15).
DD	Ballistic wind direction in hundreds of mils (01 to 64) (00 for calm wind).
FF	Ballistic wind speed in knots (00 to 99). When wind speed equals or exceeds 100 knots, add 80 to the line number.
TTT	Ballistic temperature in percent of ICAO standard to nearest 0.1 percent (000 to 999), drop hundreds digit.
RRR	Ballistic density in percent of ICAO standard to the nearest 0.1 percent (drop hundreds digit).
9	Message terminator (Punch output).
/	Missing data indicator.

e. Tables:

TABLE 1. - SPECIAL FUNCTION KEYS

F_o	START	-	Tells operator to press PIBAL Key.
F_1	CONT	-	Tells operator to press PIBAL Key.
F_2	LAST FLT	-	Tells operator to press PIBAL Key.
F_3	TAC FIRE	-	Tells operator to press PIBAL Key.
F_4	P. TAPE	-	Tells operator to press PIBAL Key.
F_5	PIBAL	-	Starts PIBAL Program.
F_6	INFO	-	Tells operator to press PIBAL Key.
F_7	OUTPUT	-	Activates whenever operator presses OUTPUT key. When pressed, LED display prompts the operator to turn on the tape perforator, if punched copy of all meteorological messages is desired. The computer outputs printed hard copies of all meteorological messages of The LAST FLIGHT ENTERED.

F_8 - DELETE - Activates after operator presses NO or STOP key. Deletes punch/print of all unwanted meteorological messages.

F_9 - NO - Used as verifier key, causes repeat of data input sequence.

F_{10} - YES - Used as verifier key, entered data are saved, and printed for the flight record.

F_{11} - DATAIN - Enters data into calculator.

TABLE 2. OCTANT OF GLOBE WHERE STATION IS LOCATED

0	0° - 90°	West Longitude	North Latitude
1	90° - 180°	"	"
2	180° - 90°	East Longitude	"
3	90° - 0°	"	"
4	Not used		
5	0° - 90°	West Longitude	South Latitude
6	90° - 180°	"	"
7	180° - 90°	East Longitude	"
8	90° - 0°	"	"
9	Coded location indicator		

TABLE 3. BAUDOT CODE (TTY-76)

Binary	Letters	Figures	Binary	Letters	Figures
0	Null	Null	16	E	3
1	T	5	17	Z	"
2	CR	CR	18	D	$
3	O	9	19	B	?
4	Space	Space	20	S	Bell
5	H	#	21	Y	6
6	N	,	22	F	!
7	M	.	23	X	/
8	LF	LF	24	A	-
9	L	)	25	W	2
10	R	4	26	J	'
11	G	&	27	FI	FI
12	I	8	28	U	7
13	P	Ø	29	Q	1
14	C	:	30	K	(
15	V	;	31	LE	LE

CR - Carriage return; LF - Line feed; FI - Figures; LE - Letters.

TABLE 4. ZONE STRUCTURE

Zone structure of the NATO, computer, and fallout metro messages.

Height Meters	Line numbers		
	NATO	Computer	Fallout
Surface	0	0	0
200	1	1	1
500	2	2	
1000	3	3	
1500	4	4	
2000	5	5	
2500	6	6	2
3000		7	
3500	7	8	
4000		9	
4500	8	10	3
5000		11	
6000	9	12	
7000	10	13	4
8000		14	
9000	11	15	5
10000		16	
11000	12	17	6
12000		18	
13000	13	19	7
14000		20	
15000	14	21	8
16000		22	
17000	15	23	9
18000		24	
19000		25	10
20000		26	
* * *			* * *
30000			15

TABLE 5. STANDARD BALLISTIC ZONE DENSITIES

Zone No.	
	0 1225.0
	1 1213.3
	2 1184.4
	3 1139.2
	4 1084.6
	5 1032.0
	6 957.0
	7 863.4
	8 777.0
	9 697.4
	10 590.0
	11 467.0
	12 364.8
	13 266.6
	14 194.8
	15 142.3

TABLE 6. STANDARD BALLISTIC ZONE TEMPERATURES

Zone No.	0	288.2
	1	287.5
	2	285.9
	3	283.3
	4	280.0
	5	276.8
	6	271.9
	7	265.5
	8	259.0
	9	252.5
	10	242.7
	11	229.8
	12	216.8
	13	216.7
	14	216.7
	15	216.7

TABLE 7. TEMPERATURE WEIGHTING FACTORS (TYPE-3 MESSAGE) (SURFACE-TO-SURFACE TRAJECTORIES)

T3

Line Number	Zone No. 1	2	3	4	5	6	7	8	9	10-15
1	1.00									
2	.27	.73								
3	.13	.20	.67							
4	.08	.12	.25	.55						
5	.05	.10	.20	.21	.44					
6	.04	.04	.09	.11	.13	.59				
7	.02	.04	.07	.09	.11	.26	.41			
8	.01	.03	.05	.04	.10	.19	.23	.35		
9-15	.01	.01	.02	.03	.03	.09	.13	.24	.44	.00

TABLE 8. DENSITY WEIGHTING FACTORS (TYPE-3 MESSAGE) (SURFACE-TO-SURFACE TRAJECTORIES)

D3

Line Number	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.43	.57													
3	.22	.31	.47												
4	.15	.21	.32	.32											
5	.11	.17	.25	.22	.25										
6	.08	.11	.17	.17	.15	.32									
7	.06	.08	.14	.13	.12	.22	.25								
8	.05	.06	.11	.11	.10	.19	.17	.21							
9	.04	.06	.09	.09	.08	.17	.15	.14	.18						
10	.03	.04	.07	.07	.07	.13	.12	.11	.11	.25					
11	.01	.03	.05	.05	.06	.12	.11	.09	.09	.16	.23				
12	.02	.03	.05	.05	.05	.11	.10	.09	.08	.14	.12	.16			
13	.02	.02	.04	.05	.05	.11	.09	.09	.08	.14	.10	.09	.12		
14	.02	.03	.05	.05	.05	.10	.09	.08	.07	.13	.11	.08	.06	.08	
15	.02	.04	.05	.05	.05	.10	.09	.08	.07	.12	.09	.08	.05	.05	.06

TABLE 9. WIND WEIGHTING FACTORS (TYPE-3 MESSAGE)

WW

Line No.	Zone No.														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.20	.80													
3	.09	.19	.72												
4	.06	.12	.26	.56											
5	.04	.08	.15	.20	.53										
6	.03	.05	.08	.09	.12	.63									
7	.02	.03	.07	.07	.08	.20	.53								
8	.02	.02	.06	.06	.06	.14	.19	.45							
9	.02	.02	.05	.05	.05	.12	.13	.20	.36						
10	.01	.02	.02	.04	.03	.07	.08	.09	.09	.55					
11	.00	.00	.01	.04	.03	.08	.08	.09	.09	.20	.38				
12	.00	.01	.01	.02	.04	.07	.07	.07	.08	.17	.16	.30			
13	.00	.01	.01	.01	.03	.07	.07	.07	.07	.15	.14	.13	.24		
14	.00	.01	.01	.01	.02	.07	.07	.07	.07	.13	.13	.13	.10	.18	
15	.00	.01	.01	.01	.02	.07	.07	.07	.07	.12	.12	.11	.10	.08	.14

TABLE 10. TEMPERATURE WEIGHTING FACTORS (TYPE-2 MESSAGE)

Line No.	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13-15
1	1.00												
2	.63	.37											
3	.37	.37	.26										
4	.25	.30	.35	.10									
5	.20	.24	.30	.18	.08								
6	.13	.19	.24	.18	.14	.12							
7	.10	.14	.20	.16	.14	.19	.07						
8	.09	.10	.17	.15	.13	.20	.12	.04					
9	.07	.09	.14	.13	.12	.19	.15	.08	.03				
10	.05	.08	.12	.10	.10	.17	.14	.10	.08	.06			
11	.05	.06	.10	.09	.08	.15	.13	.12	.10	.12	.00		
12	.04	.06	.10	.08	.08	.14	.13	.11	.10	.16	.00	.00	
13-15	.05	.06	.10	.09	.08	.16	.12	.13	.11	.10	.00	.00	.00

TABLE 11. DENSITY WEIGHTING FACTORS (TYPE-2 MESSAGE) (SURFACE-TO-AIR TRAJECTORIES)

Line No.	Zone No.														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.63	.37													
3	.37	.37	.26												
4	.25	.30	.35	.10											
5	.20	.24	.30	.18	.08										
6	.13	.19	.24	.18	.14	.12									
7	.10	.14	.20	.16	.14	.19	.07								
8	.09	.10	.17	.15	.13	.20	.12	.04							
9	.07	.09	.14	.13	.12	.19	.15	.08	.03						
10	.05	.08	.12	.10	.10	.17	.14	.10	.08	.06					
11	.04	.06	.10	.08	.08	.15	.13	.10	.10	.12	.04				
12	.04	.06	.09	.08	.08	.13	.12	.10	.08	.13	.07	.02			
13	.03	.05	.08	.08	.06	.12	.11	.10	.08	.13	.09	.05	.02		
14	.03	.05	.06	.07	.07	.11	.10	.09	.08	.13	.10	.06	.04	.01	
15	.02	.05	.06	.07	.05	.11	.10	.08	.08	.13	.10	.07	.05	.03	.00

TABLE 12. WIND WEIGHTING FACTORS (TYPE-2 MESSAGE)

WW

Line No.	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.50	.50													
3	.29	.33	.38												
4	.18	.23	.39	.20											
5	.13	.18	.31	.27	.11										
6	.08	.12	.22	.20	.19	.19									
7	.07	.08	.16	.15	.16	.27	.11								
8	.04	.08	.13	.12	.13	.24	.18	.08							
9	.04	.06	.11	.10	.10	.21	.20	.12	.06						
10	.03	.04	.08	.08	.08	.16	.15	.14	.13	.11					
11	.02	.04	.06	.07	.06	.13	.13	.12	.11	.18	.08				
12	.03	.04	.07	.07	.07	.12	.11	.10	.08	.15	.10	.06			
13	.02	.04	.05	.06	.06	.11	.10	.09	.08	.14	.11	.09	.05		
14	.02	.04	.05	.06	.04	.09	.09	.09	.08	.13	.11	.09	.06	.05	
15	.01	.03	.05	.04	.05	.09	.09	.08	.07	.12	.10	.09	.08	.06	.04

2. FILE DESCRIPTION

The Hewlett-Packard cartridge has two tracks of 125K bytes for recording programs and data. The titles on these tracks and size of these files follow:

a. Track Ø

		Section
File Ø:	Master Supervisory File	Command supervisor
File 1:	Functional Key Definitions	
File 2:	Master Data File Library	
Files 3 to 65 inc:	Departure Tables	Data

b. Track 1

		Section
File Ø:	PIBAL Program	PIBAL
File 1:	P$, Q$, O$, L$, N$, G$, D$	
File 2:	R(*), Y(*)	
File 3:	H(*), D(*)	
File 4:	Sound Ranging, Computer meteorology, fallout, ballistic 3	Output
File 5:	Ballistic 2	
File 6:	Departure Table Input Program	
File 7 thru 40:	Departure Tables, Z$	Data

c. T List Files

The following is a t list of the files on tracks Ø and 1 showing the size of the files and the amount of bytes stored in each file.

T LIST TRACK Ø

trk Ø	Bytes	Size
#0		
6	664	1000
#1		
5	682	1000
#2		
3	2106	2106
#3 thru #65		
3	2106	2106

T LIST TRACK 1

trk 1	Bytes	Size
#0		
6	13898	15000
#1		
3	420	500
#2		
2	2992	4000
#3		
2	1776	3000
#4		
6	13196	15000
#5		
6	6700	7000
#6		
6	2564	3000
#7 thru #40		
3	2106	2106

3. VARIABLE DESCRIPTION

a. Array Variables

dim A(12) -- Launch Values

A(1) = Azimuth offset at launch.
A(2) = Horizontal distance offset.
A(3) = Surface wind direction.
A(4) = Surface wind speed.
A(5) = Balloon weight.
A(6) = Surface pressure.
A(7) = Surface temperature.
A(8) = Surface density.
A(9) = Virtual temperature.
A(10) = Saturation vapor pressure at dry bulb temperature.
A(11) = Surface relative humidity.
A(12) = Actual vapor pressure.

dim B(15) - Sound Ranging Values

B(1) = Zone 1 wind speed.
B(2) = Zone 2 wind speed.
B(3) = Zone 3 wind speed.
B(4) = Zone 4 wind speed.
B(5) = Zone 1 wind direction.
B(6) = Zone 2 wind direction.
B(7) = Zone 3 wind direction.
B(8) = Zone 4 wind direction.
B(9) = Surface wind speed.
B(10) = Surface wind direction.
B(11) thru B(15) not used.

dim C(0:16, 1:14) - Ballistic Weights and Values
0:16 - zones where 0 = surface
1-15 = lines

C(N,1) = Zone level number.
C(N,2) = Wind speed.
C(N,3) = Temperature.
C(N,4) = Pressure.
C(N,5) = Wind azimuth.
C(N,6) = % of standard temperature.
C(N,7) = Weighted ballistic 3 temperature.
C(N,8) = Weighted ballistic 2 temperature.
C(N,9) = Densities.
C(N,10) = % of standard density.
C(N,11) = Weighted ballistic 3 densities.
C(N,12) = Weighted ballistic 2 densities.
C(N,13) = Weighted ballistic 3 or 2 X wind component.
C(N,14) = Weighted ballistic 3 or 2 Y wind component.

dim D(0:29) - Zone Heights in OUTPUT Section

dim F(-1:60, 1:6) - Significant Level Data

at -1 level

#1	#2	#3	#4	#5	#6
Azimuth offset	Horizontal distance	First Missing Mandatory	Second Missing Mandatory	Third Missing Mandatory	Fourth Missing Mandatory

at 0 level
1-60 levels

#1	#2	#3	#4	#5	#6
Temperature ^{o}C	Humidity %	Temp virt ^{o}K	Pressure mb	Time min	Geopotential height meters

dim H(-2:45, 1:6) - Zone Height Data

at -2 level

#1	#2	#3	#4	#5	#6
		Counter		SFC Wind Direction	SFC Wind Speed

at -1 level

#1	#2	#3	#4	#5	#6
	SFC pressure		SFC temp Tv ^{o}K	Azimuth Offset	Horizontal distance

at 0-45 levels

#1	#2	#3	#4	#5	#6
Geometric height	log press	miscellaneous	top of zone time	top of zone azimuth angle	top of zone elevation angle

Miscellaneous

H(0,3) = Acceleration of Gravity Ratio Constant.
H(1,3) = P = zone level.
H(2,3) = T = significant level.
H(3,3) = Missing angles indicator.
H(4,3) = Angle time interval (0.1, 0.5, 1.0).
H(5,3) = Last correct time of angles.
H(6,3) = Last correct elevation angle.
H(7,3) = Last correct azimuth angle.
H(8,3) = Time below zone.
H(9,3) = Elevation angle below zone.
H(10,3) = Azimuth angle below zone.
H(11,3) = Time above zone.
H(12,3) = Elevation angle above zone.
H(13,3) = Azimuth angle above zone.
H(14,3) = Reference time below zone.
H(15,3) = Reference time above zone.

b. String Variables

A$, B$, C$ - string areas.
D$(16) - date of flight.

Date-Time Group Format:

DDbTTTTZbMMMbYYb

DD - numerical day of month (01 to 31).
TTTT - time value (0001 to 2400).
MMM - alphabetic designation of month (JAN-DEC).
YY - year (00 - 99).
b - blank.

E$ (16) - string array for linearity.
F$ (15) - wind data strings.
G$ (192) - recorder check linearity data.
H$ (2,32) - Baudot conversion array.
L$ (16) - recorder check date-time group.
N$ (6) - recorder serial number.
O$ (16) - previous survey date-time group.
Z$ (10,208) - computer met data.

dim P$(102) - Header Array

	1	2	3	4	5	6	7	8	9	10	11	12		
P$ (1,12) =	M	E	T	C	M	Q	A	A	A	0	0	0	)	Computer met message
P$ (13,24)	Y	Y	G	G	G	V	H	H	H	P	P	P	)	
P$ (25,36)	M	E	T	B	3	Q	A	A	A	0	0	0		NATO 3
P$ (37,48)	M	E	T	B	2	Q	A	A	A	0	0	0		NATO 2
P$ (49,60)	M	E	T	F	M	Q	A	A	A	0	0	0		Fallout
P$ (61,72)	Y	Y	G	G	G	H	H	H	H	T	R	0		
P$ (73,84)	M	E	T	S	R	Q	A	A	A	0	0	0		Sound Ranging
P$ (85,99)	Y	Y	G	G	G	V	+	T	T	T	D	D	D	F	F
P$ (100,100)	NOT USED.													
P$ (101,101)	NOT USED.													
P$ (102,102)	NOT USED.													

c. r-variables

Extensive use of r-variables r1 through r15 is made in the ballistic messages. These variables hold the weighting factors that are applied to density, temperature, and wind components. They represent the zone levels 1 to 15.

4. FLOW CHARTS

This section provides flow charts showing the relationship between the individual routines and the program flow.

a. System Program

The system logic diagrams are shown in figures 1-3.

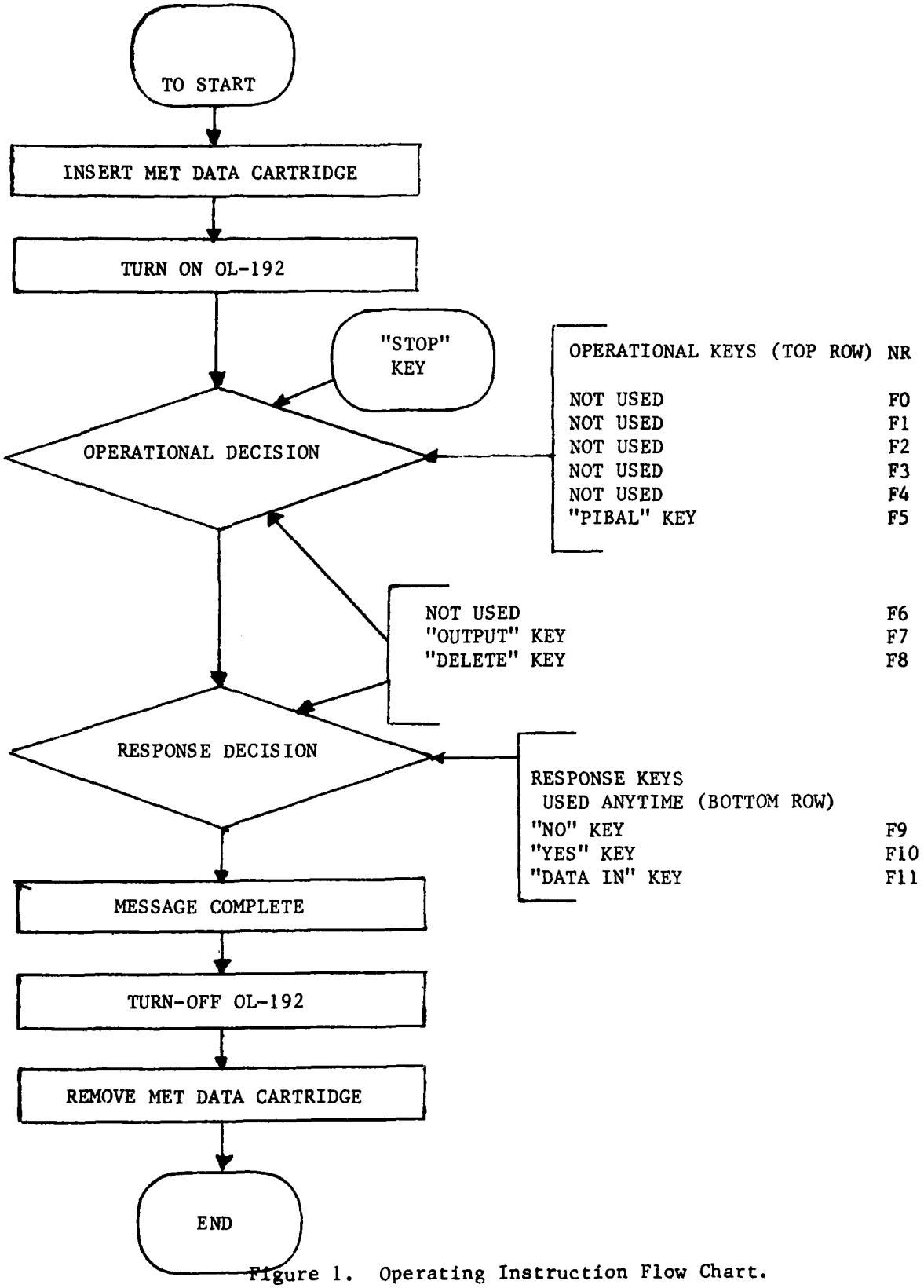


Figure 1. Operating Instruction Flow Chart.

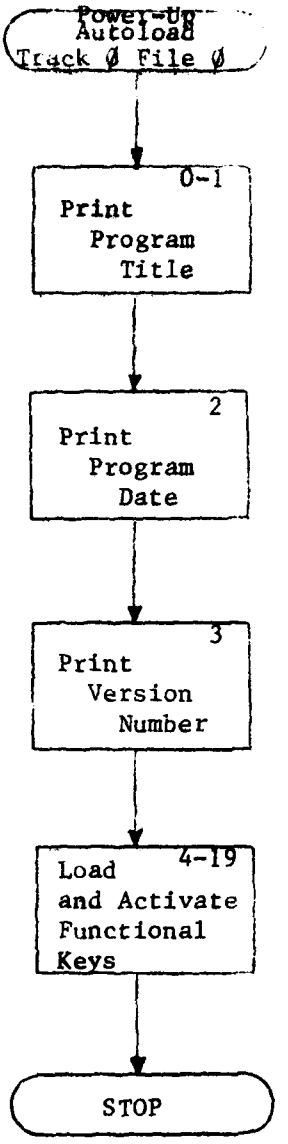


Figure 2. Autoload and Master Supervisory File
(Track 0 File 0)

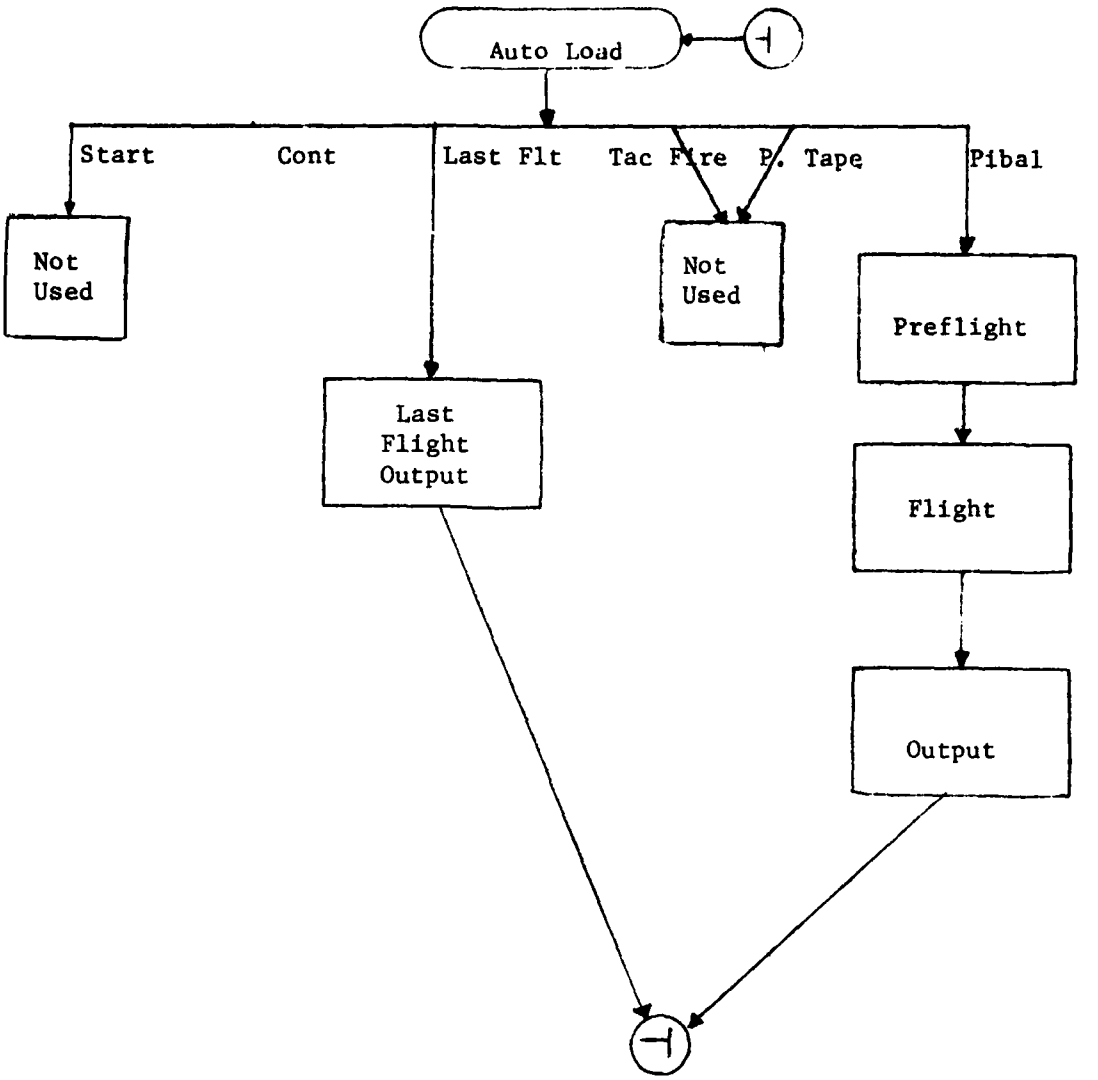


Figure 3. Functional Keys and Program Flow.

b. Pibal Program

In this portion the preflight, flight, and output routine flow charts are shown in figures 4a - 4g.

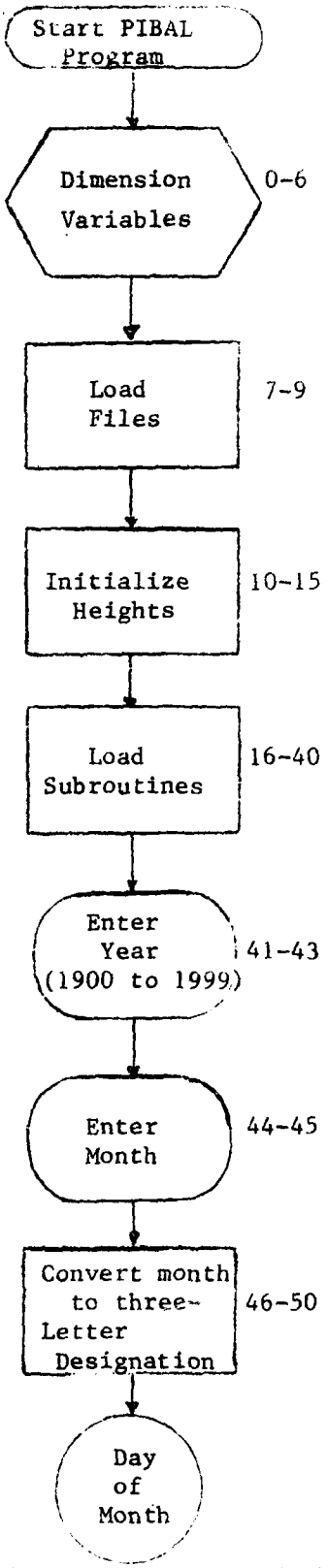


Figure 4a. PIBAL (Track 1 File 0) (1 of 7).

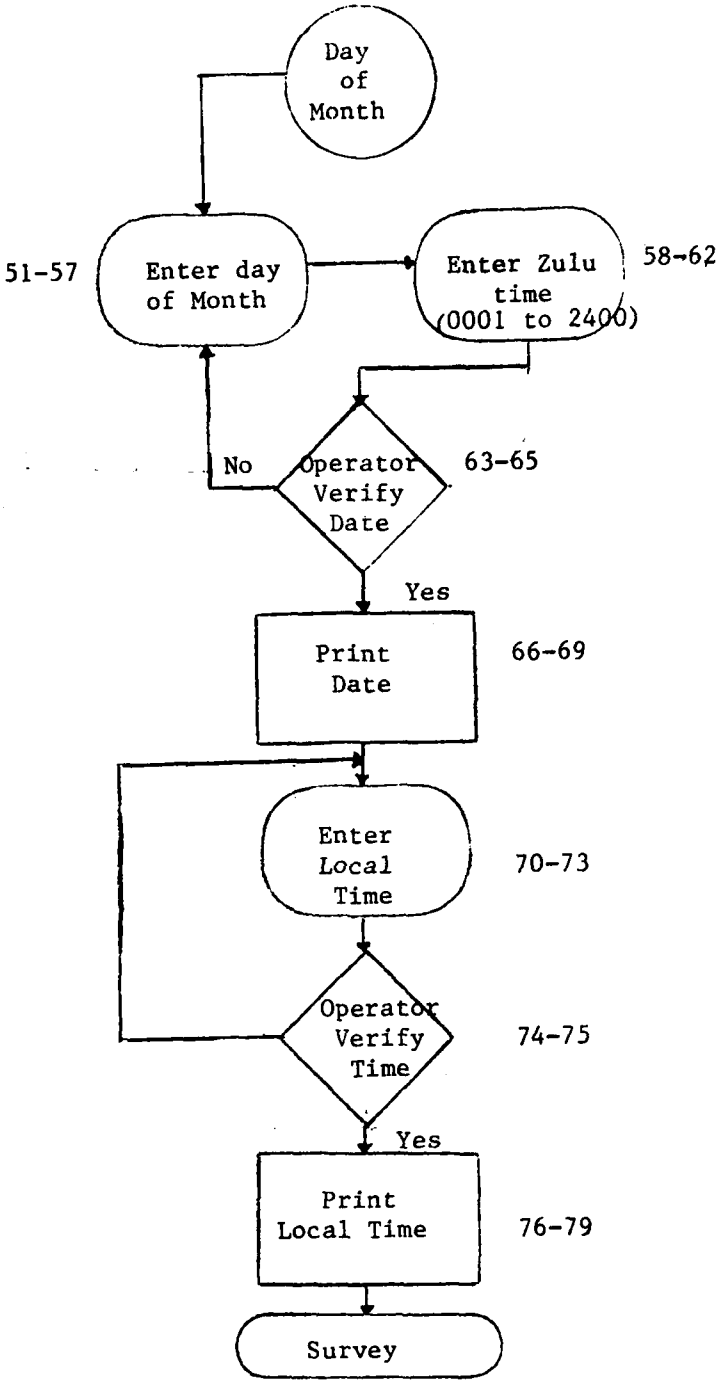


Figure 4b. PIBAL - continued (2 of 7).

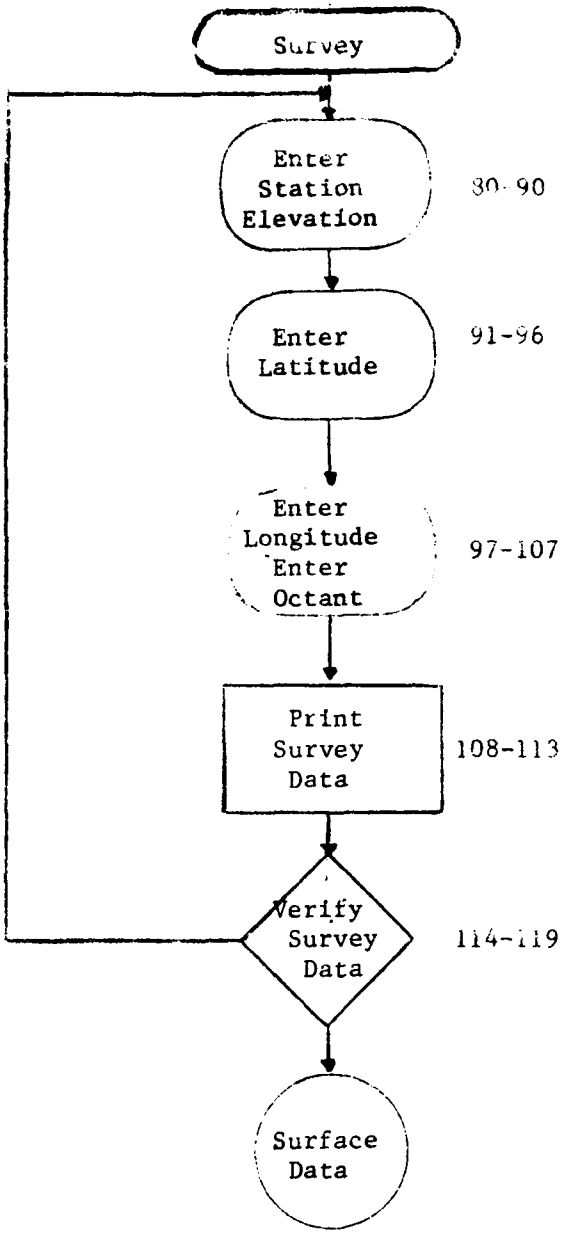


Figure 4c. PIBAL - continued (Survey) (3 of 7).

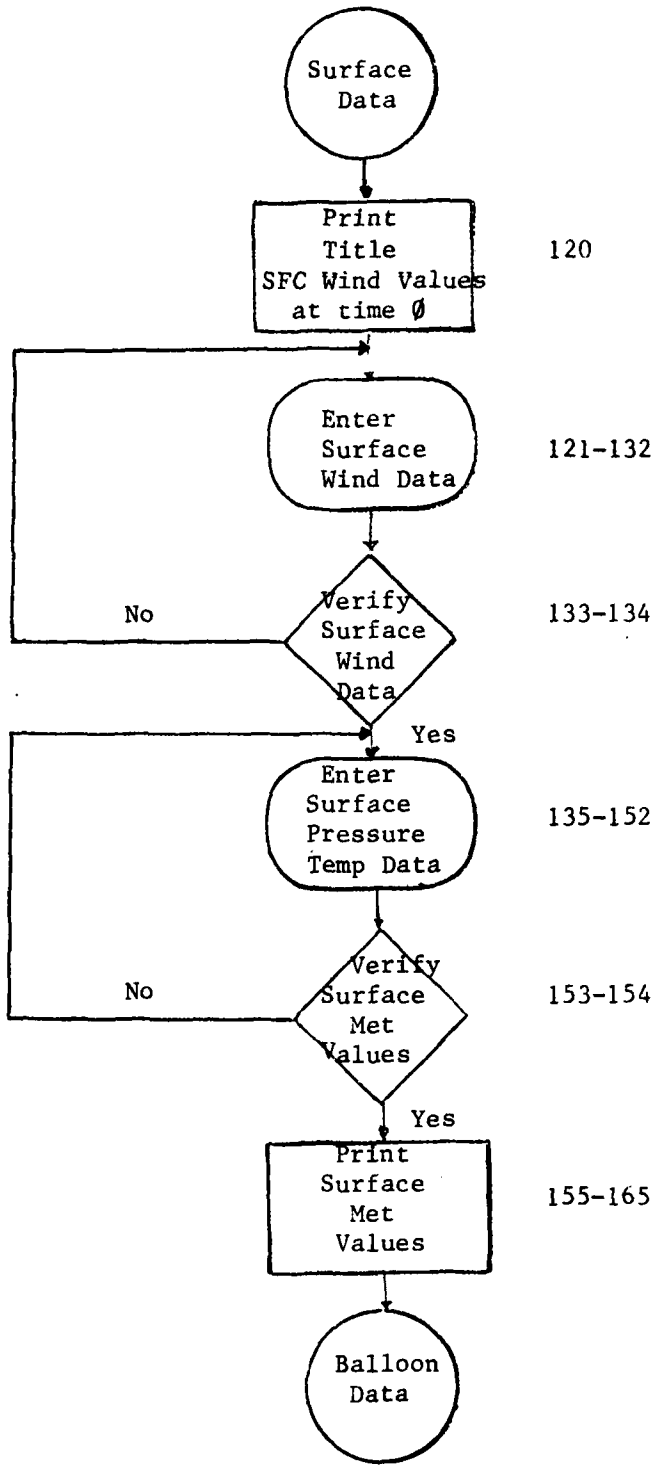


Figure 4d. PIBAL - continued (Surface Data) (4 of 7).

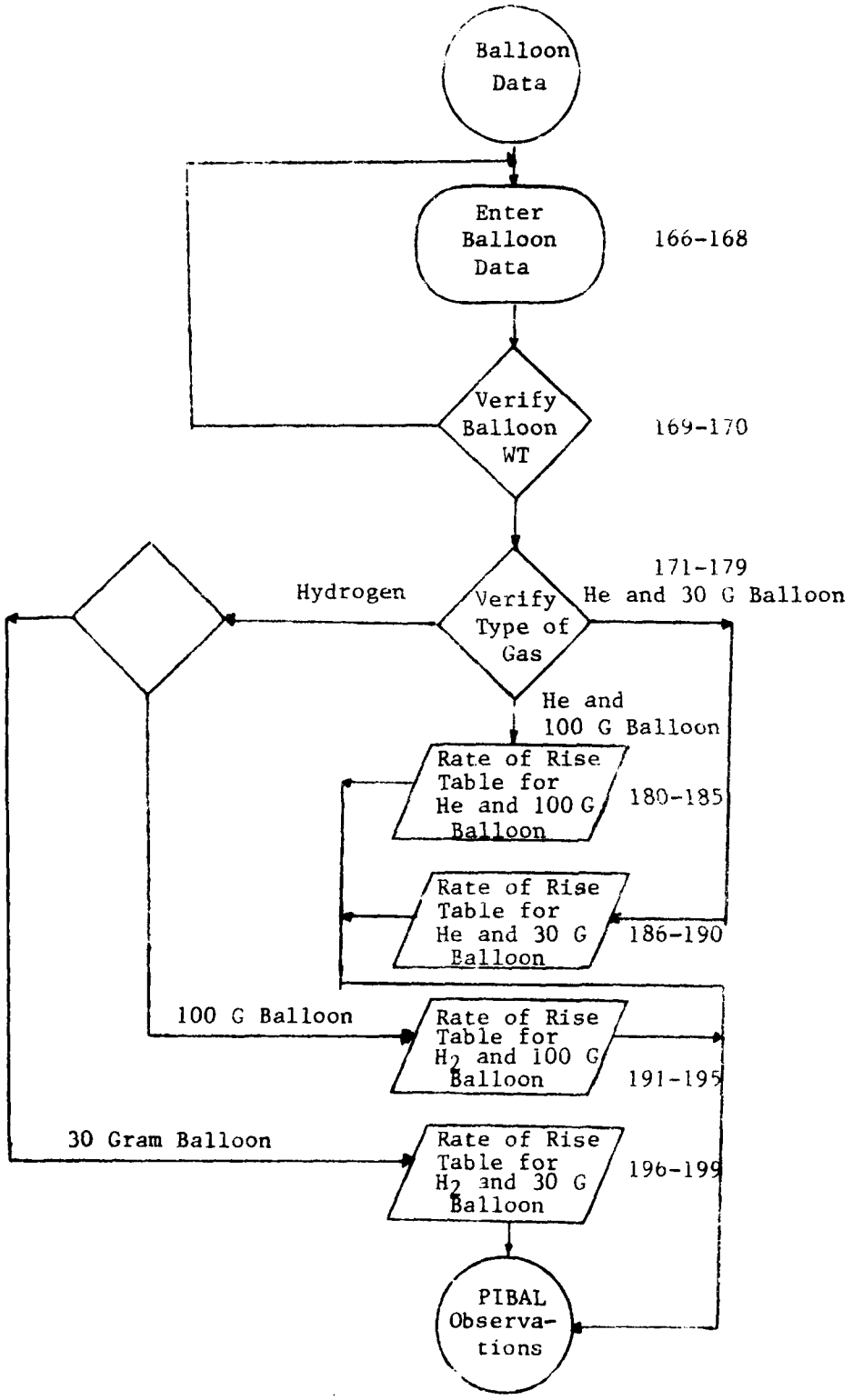


Figure 4e. PIBAL - continued (Balloon Data) (5 of 7).

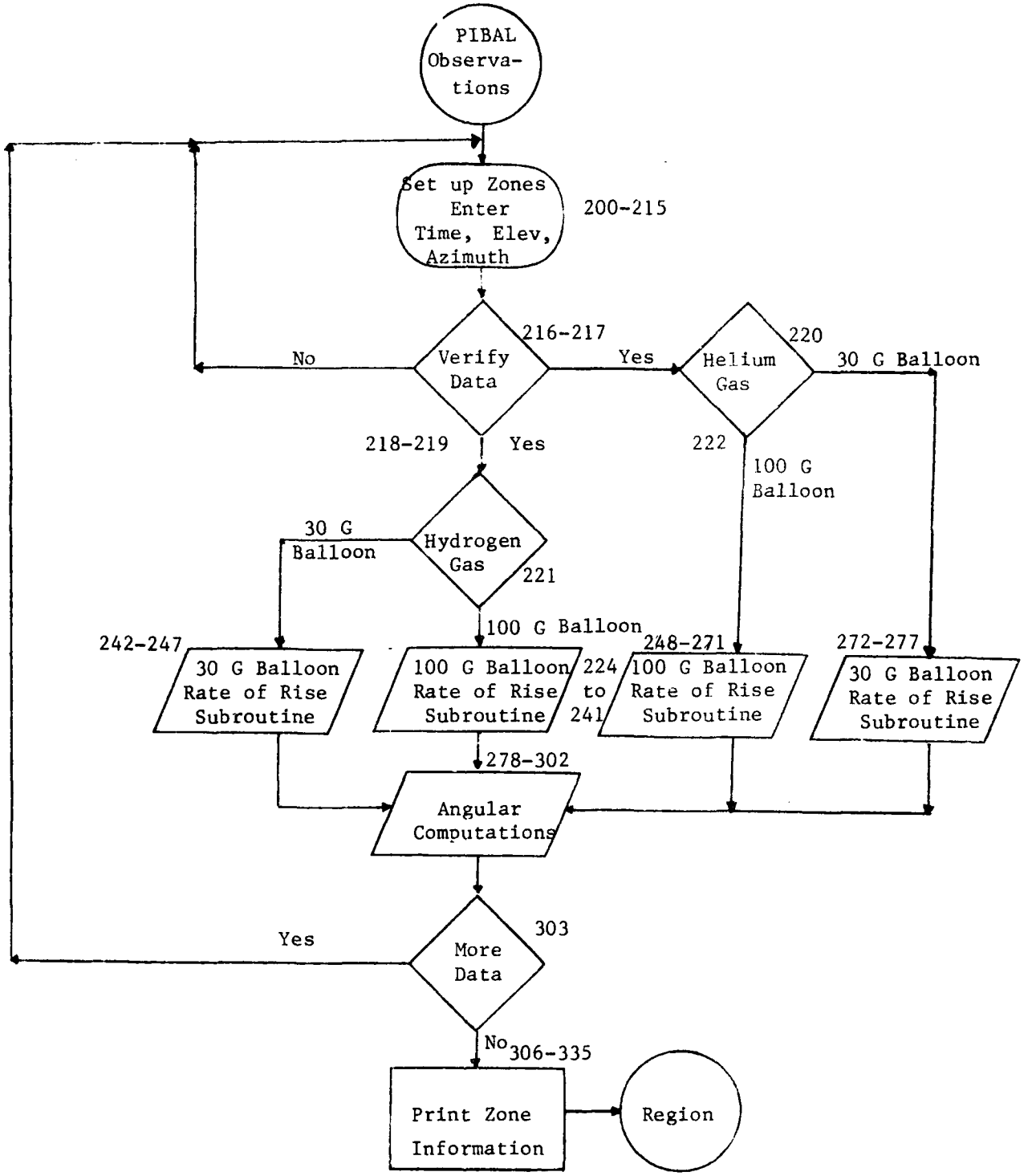


Figure 4f. PIBAL - Continued (PIBAL Observations) (6 of 7).

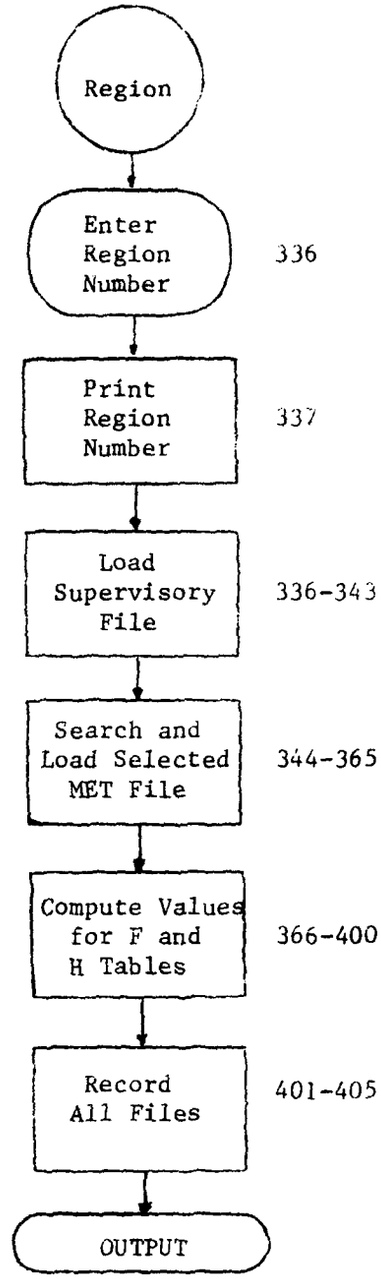


Figure 4g. PIBAL - Continued (Computer MET for Regions) (7 of 7).

c. Output Program

The output program flow charts for the artillery met messages are shown in figures 5a - 5e.

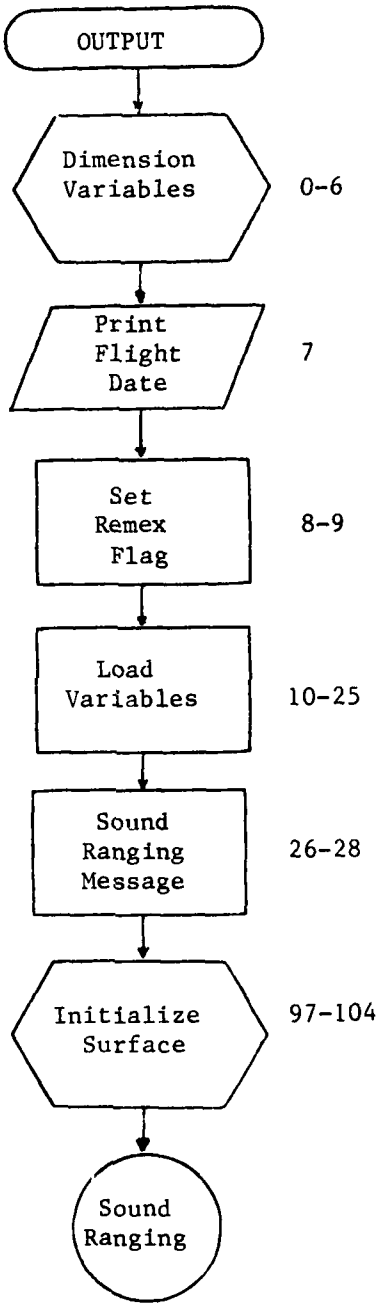


Figure 5a. Output Sound Ranging, Fallout, Ballistic MET 3 (Track 1 File 4) (1 of 5).

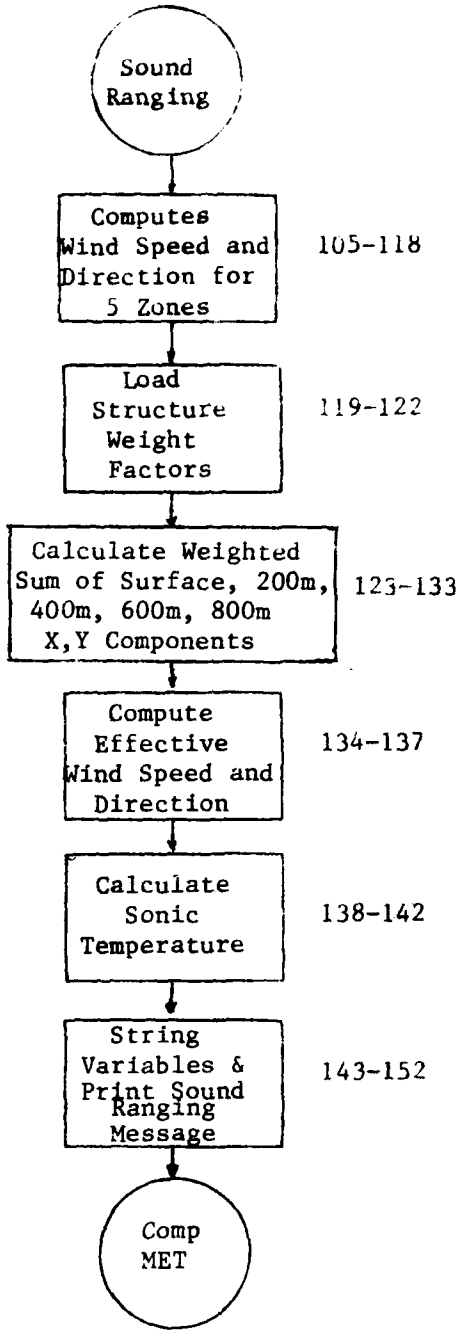


Figure 5b. Output - Continued (2 of 5).

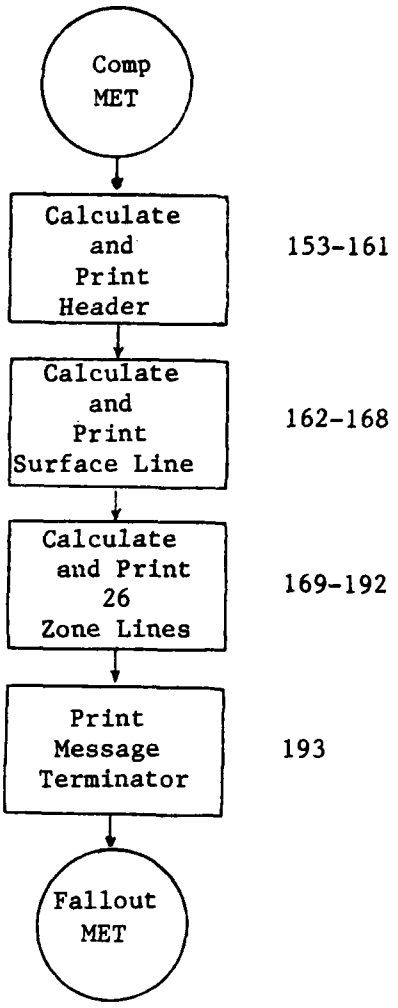


Figure 5c. Output - continued (3 of 5)

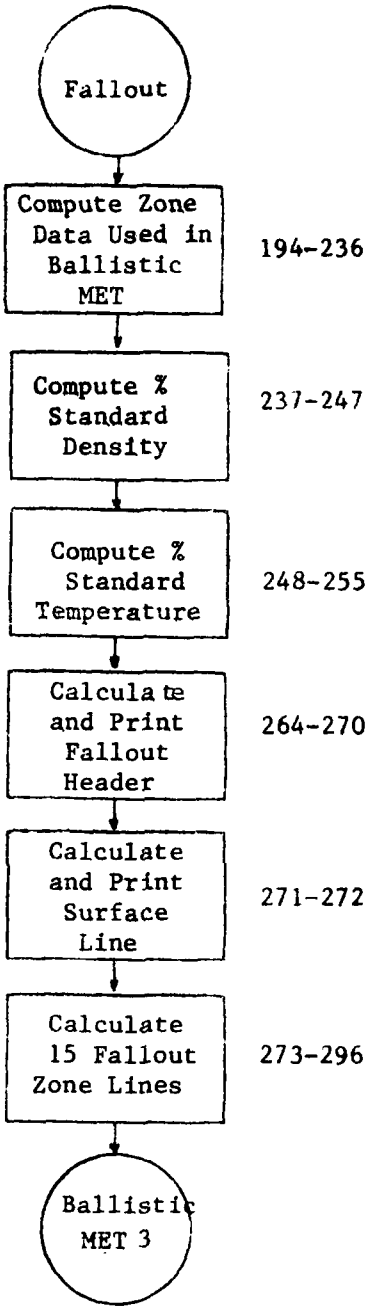


Figure 5d. Output - continued (4 of 5)

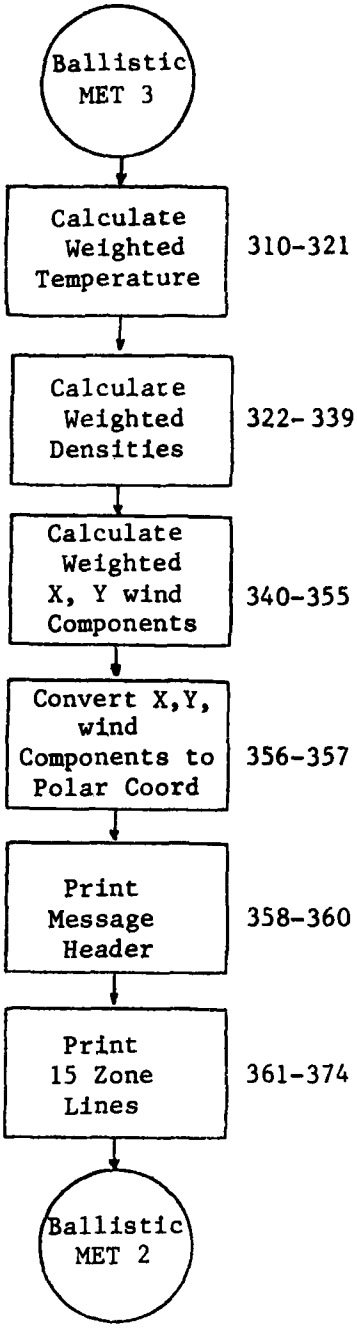


Figure 5e. Output - continued (5 of 5)

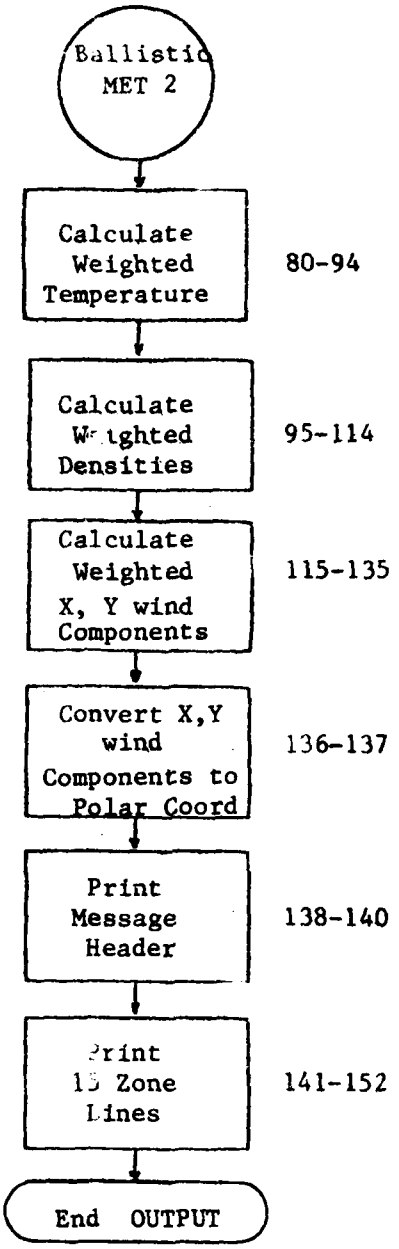


Figure 6. Output Ballistic MET 2 (Track 1 File 5)

d. Departure Table Input Program

The program for the Departure Table input is shown in figure 7.

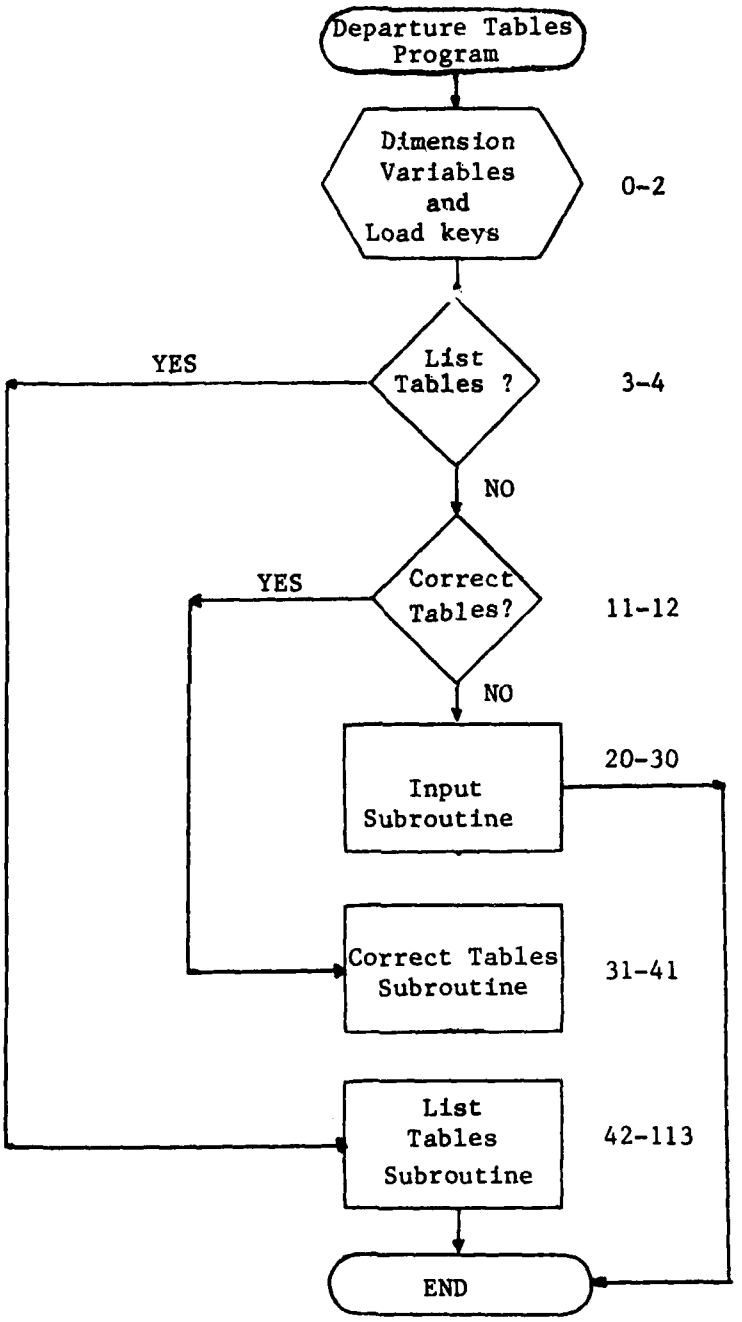


Figure 7. Departure Table Input Program (Track 1 File 6)

5. PROGRAM LISTING.

a. System Listing.

(1) Master Supervisory File

```
TRACK 0 FILE 0


0: prt "OL-192 ARTILLERYMETEOROLOGY....."
1: prt ".PIBAL PROGRAM.."
2: prt "DATED:23 JULY 79";spc 2
3: prt "TAPE NO.  P1-000"
4: trk 0;ldk 1
5: dsp "USE FUNCTIONAL KEYS TO ACTIVATE";gsb "BBB"
6: stp
7: dsp "PLEASE USE UPPER ROW KEYS";gsb "BBB"
8: stp
9: dsp "LET'S TRY IT AGAIN";gsb "BBB"
10: gto -6
11: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
12: " THIS PROGRAM REQUIRES THE FOLLOWING HARDWARE":
13: "     HP 9825A CALCULATOR WITH 24K RWM":
14: "           HP9825A OPERATING SYSTEM ROM 67905":
15: "           GENERAL IO/ EXTENDED IO ROM 98213A":
16: "           STRING-ADV PROGRAMMING ROM 98210A":
17: "           REMEX READER/PERFORATOR FOR PUNCHED OUTPUT":
18: "           INTERFACE CABLE 98032A OPT  A03":
19: end
*10713
```

(2) Functional Keys

```
f0=f12:     *dsp  "START- -PRESS PIBAL"
f1=f13:     *dsp  "CONTINUATION- -PRESS PIBAL"
f2=f14:     *dsp  "LAST FLIGHT- -PRESS PIBAL"
f3=f15:     *dsp  "TACFIRE- -PRESS PIBAL"
f4=f16:     *dsp  "P. TAPE- -PRESS PIBAL"
f5=f17:     *dsp  "PROCESSING PIBAL"; Trk 1; ldp0
f6=f18:     "dsp  "INFO- -PRESS PIBAL"
f7=f19:     *dsp  "PROCESSING MET MESSAGES"; Trk 1; ldp 4
f8=f20:     *dsp  "DELETE"
f9=f21:            /-2tn↑10
f10=f22:           /-1tn↑10
f11=f23:           /
```

b. PIBAL

PIBAL PROGRAM DATED 23 JULY 1979

INPUT PROGRAM ON TRACK 1 FILE 0

TRACK 1 FILE 0

```
0: sfg 0;cfg 3
1: dim A$[40],R[4],C$[32]
2: dim Z$[10,208]
3: dim P$[102],Q$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
4: dim F[-1:60,1:6],Y[2]
5: dim H[-2:29,1:6],D[0:29]
6: dim A[12]
7: trk 1;ldf 1,P$,Q$,O$,L$,N$,G$,D$
8: trk 1;ldf 2,F[*],Y[*]
9: trk 1;ldf 3,H[*],D[*]
10: fxd 0;prt "COUNTER",H[-2,3];spc 2
11: 0→H[0,1];200→H[1,1];400→H[2,1];500→H[3,1];600→H[4,1];800→H[5,1]
12: 0→D[0];200→D[1];400→D[2];500→D[3];600→D[4];800→D[5]
13: 6→I;for J=1000 to 5000 by 500;J→H[I,1]→D[I];I+1→I;next J;15→I
14: for J=6000 to 20000 by 1000;J→H[I,1]→D[I];I+1→I;next J
15: gto +26
16: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
17: "DELETE":cfg 3;dsp "DELETING ZONE INFORMATION";ret
18: "DI":fxd 1;2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep;dsp C$[1,32]
19: ent "",A$;if A$[1,1]="0" or A$[1,1]="1" or A$[1,1]="2";gto +9
20: if A$[1,1]="3" or A$[1,1]="4" or A$[1,1]="5" or A$[1,1]="6";gto +8
21: if A$[1,1]="7" or A$[1,1]="8" or A$[1,1]="9" or A$[1,2]=".0";gto +7
22: if A$[1,2]=".1" or A$[1,2]=".2" or A$[1,2]=".3" or A$[1,2]=".4";gto +6
23: if A$[1,2]=".5" or A$[1,2]=".6" or A$[1,2]=".7" or A$[1,2]=".8";gto +5
24: if A$[1,2]=".9";gto +4
25: if A$[1,7]="*";gto +279
26: stp
27: gto -9
28: if A#2;gto -10
29: val(A$)→L;if L<0;gto -11
30: "                                    "→C$[1,32]
31: ret
32: "Y/N":2→A;beep;dsp C$[1,32];ent "",A
33: if A#-2tn^10 and A#-1tn^10;gto -1
34: "                                    "→C$[1,32]
35: ret
36: "R→P":√(XX+YY)→R
37: atn(Y/(X+1e-90*(X=0)))+2sgn(Y)*atn(1e99)*(X<0)→A
38: if A<0;360+A→A
39: ret
40: "P→R":Rcos(A)→X;Rsin(A)→Y;ret
41: "ENTER YEAR (1900 to 1999)"→C$;gsb "DI"
42: if L<1900 or L>1999;gto -1
43: val(A$[3,4])→R[1];A$[3,4]→D$[14,15]
44: "ENTER NUMBER OF MONTH (1 to 12)"→C$;gsb "DI"
 5: L→R[2];if R[2]<1 or R[2]>12;gto -1
 6: "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"→A$
47: A$[3*R[2]-2,3*R[2]]→D$[10,12]
48: 31→N;if R[2]=2;28→N;if int(R[1]/4)=R[1]/4;29→N
49: if R[2]=4 or R[2]=6 or R[2]=9 or R[2]=11;30→N
*26420
```

```
50: str(N)→A$;A$[2]→A$[1]
51: "ENTER DAY OF MONTH (1 to "→C$[1,25];A$[1,2]→C$[26,27]
52: ")"→C$[28];gsb "DI"
53: L→R[3];if R[3]<1 or R[3]>N;gto -3
54: if R[3]>9;str(R[3])→D$[1,3];gto +2
55: str(R[3])→D$[2,3];"0"→D$[2,2]
56: D$[2,3]→D$[1,2];""→D$[3,3]
57: 2→A;"0"→A$;beep;wait 150;beep
58: ent "ENTER ZULU TIME (0001 to 2400)",A$
59: if A#2 or len(A$)#4;gto -3
60: val(A$)→R[4];if R[4]<1 or R[4]>2400;gto -4
61: if frc(R[4]/100)>.59;gto -5
62: A$[1,4]→D$[4,7];"Z"→D$[8,8]
63: "IS THE DATE "→C$[1,12];D$→C$[13,29];"?"→C$[30];gsb "Y/N"
64: if A=-2tn^10;gto -23
65: dsp D$[1,16];gsb "BBB"
66: prt "DATE:",D$;spc 2
67: D$[1,2]→P$[13,14];D$[4,5]→P$[15,16];"00"→P$[17,18]
68: int(val(D$[6,7])/6)→A;str(A)→A$;A$[2,2]→P$[17,17]
69: 2→A;"0"→A$;beep;wait 150;beep
70: ent "ENTER LOCAL TIME(0001 to 2400)",A$
71: if A#2 or len(A$)#4;gto -3
72: val(A$)→Q;if Q<1 or Q>2400;gto -4
73: if frc(Q/100)>.59;gto -5
74: "IS THE TIME "→C$[1,12];A$[1,4]→C$[13,16];"?"→C$[17];gsb "Y/N"
75: if A=-2tn^10;gto -6
76: dsp C$[1,16];gsb "BBB"
77: prt "LOCAL TIME:",A$[1,4];spc 2
78: D$[1,16]→O$[1,16]
79: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
80: dsp "PERFORM SURVEY CONTROL";gsb "BBB"
81: ent "ENTER STATION ELEVATION (meters)",A$
82: if A#2;gto -3
83: if val(A$)<-200 or val(A$)>=450;gto -4
84: int(val(A$[1,4])/10+.5)→K;fxd 0
85: if K>=100;str(K)→P$[18,21];"0"→P$[18,18];gto +6
86: if K>=10;str(K)→P$[19,21];"00"→P$[18,19];gto +5
87: if K>=0;str(K)→P$[20,21];"000"→P$[18,20];gto +4
88: int(abs(val(A$[1,4])/10-.5))→K;"0"→P$[18,19]
89: if K>=10;str(K)→P$[19,21];"0-"→P$[18,19];gto +2
90: if K>=0;str(K)→P$[20,21];"0-0"→P$[18,20]
91: "ENTER LATITUDE (0.0 to 90.0)"→C$;gsb "DI"
92: if L<0 or L>90;gto -1
93: L→K;str(K)→A$[1,5]
94: if A=-2tn^10;gto -3
95: A$[2,3]→P$[7,8];A$[5,5]→P$[9,9]
96: if K<10;"0"→P$[7,7];A$[2,2]→P$[8,8];A$[4,4]→P$[9,9]
97: "ENTER LONGITUDE (0.0 to 99.9)"→C$;gsb "DI"
98: if L<0 or L>99.9;gto -1
99: L→K;str(K)→A$[1,5]
*23345
```

```
100: if A=-2tn^10;gto -3
101: A$[2,3]→P$[10,11];A$[5,5]→P$[12,12]
102: if K<10;"0"→P$[10,10];A$[2,2]→P$[11,11];A$[4,4]→P$[12,12]
103: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
104: ent "ENTER OCTANT (0 to 8 not 4)",A$
105: if A#2 or len(A$)#1;gto -2
106: if val(A$)<0 or val(A$)=4 or val(A$)>8;gto -3
107: A$[1,1]→P$[6,6]
108: dsp "SURVEY DATA";gsb "BBB"
109: prt "SURVEY DATED:",O$
110: fxd 0;prt "OCTANT:",val(P$[6,6])
111: fxd 1;prt "LATITUDE:",val(P$[7,9])/10
112: fxd 1;prt "LONGITUDE:",val(P$[10,12])/10
113: fxd 0;val(P$[19,21])*10→K;prt "ELEVATION:",K
114: "IS SURVEY DATA CORRECT?"→C$[1,32];gsb "Y/N"
115: if A=-2tn^10;prt "---REPEATING---";gto -36
116: P$[1,12]→P$[25,36]→P$[37,48]→P$[49,60];"B3"→P$[28,29];"B2"→P$[40,41]
117: "FM"→P$[52,53];"CM"→P$[4,5];P$[13,24]→P$[61,72];"TRO"→P$[70,71]
118: P$[1,12]→P$[73,84];"SR"→P$[76,77]
119: P$[13,24]→P$[85,96];"+TTTDDDFF"→P$[91,99]
120: spc 1;prt "SFC WIND VALUES AT TIME ""0"":"
121: "ENTER AZIMUTH OFFSET AT LAUNCH "→C$;gsb "DI"
122: L→A[1];if L<=0 or L>360;gto -1
123: prt " AZ OFFSET",A[1]
124: "ENTER HORIZONTAL DISTANCE OFFSET"→C$;gsb "DI"
125: L→A[2];if L<.1 or L>1500;gto -1
126: prt " HOR DIS",A[2]
127: "ENTER SFC WIND DIRECTION(deg.)"→C$;gsb "DI"
128: L→A[3];if L=0 or L>360;gto -1
129: prt " WIND DIR",A[3]
130: "ENTER SFC WIND SPEED"→C$;gsb "DI"
131: L→A[4];if L<0 or L>99;gto -1
132: prt " WIND SPEED",A[4];spc 2
133: "ARE PRINTED SFC WIND VALUES OK ?"→C$[1,32];gsb "Y/N"
134: if A=-2tn^10;prt "---REPEATING----";gto -13
135: "ENTER SURFACE PRESS AT LAUNCH"→C$;gsb "DI"
136: L→A[6];if L<650 or L>1100;gto -1
137: prt "SFC MET VALUES:"
138: fxd 1;prt "PRESSURE",A[6]
139: str(prnd(A[6],0))→A$;A$[2,4]→P$[22,24]
140: if A[6]>=1000;A$[3,5]→P$[22,24]
141: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
142: ent "ENTER SFC TEMPERATURE(degC)",A$
143: if A#2;gto -2
144: if val(A$)<-25 or val(A$)>100;gto -3
145: val(A$)→A[7]
146: prt "TEMPERATURE",A[7]
147: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
148: ent "ENTER WET BULB TEMP(degC)",A$
149: if A#2;gto -2
3233
```

```
150: if val(A$)<-25 or val(A$)>100;gto -3
151: val(A$)→A[9]
152: prt "W BULB TEMP",A[9];spc 2
153: "ARE SFC VALUES OK ?"→C$;gsb "Y/N"
154: if A=-2tn^10;prt "---REPEATING----";gto -19
155: 6.11*10^(7.5A[7]/(237.3+A[7]))→A[10]
156: 6.11*10^(7.5A[9]/(237.3+A[9]))→A[11]
157: .00066(1+.0015A[9])A[6](A[7]-A[9])→A[12]
158: A[11]-A[12]→A[12]
159: A[12]/A[10]→A[11]
160: 100A[11]→G;if G<0;0→G
161: if G>100;100→G
162: fxd 0;prt "HUM %",G
163: (A[7]+273.16)/(1-.37803A[12]/A[6])→A[9]
164: 348.38395A[6]/A[9]12.25→A[8]
165: fxd 1;prt "SFC DENSITY",A[8];spc 2
166: "ENTER BALLOON WEIGHT(gms)"→C$;gsb "DI"
167: L→A[5];if L#30 and L#100;gto -1
168: fxd 0;prt "BALLOON WGT",A[5];spc 2
169: "IS BALLOON WEIGHT OK ?"→C$[1,32];gsb "Y/N"
170: if A=-2tn^10;prt "---REPEATING----";gto -4
171: "WAS HELIUM USED FOR INFLATION?"→C$[1,32];gsb "Y/N"
172: if A=-1tn^10;prt "HELIUM"
173: if A=-2tn^10;prt "HYDROGEN"
  4: if A=-1tn^10 and A[5]=100;101→A[5]
175: if A=-1tn^10 and A[5]=30;31→A[5]
176: if A[5]=101;gto +4
177: if A[5]=31;gto +9
178: if A[5]=100;gto +13
179: if A[5]=30;gto +17
180: .571→H[1,4];1.1563→H[2,4];1.4688→H[3,4];1.7813→H[4,4];2.4194→H[5,4]
181: 3.0667→H[6,4];4.7167→H[7,4];6.4138→H[8,4];8.1579→H[9,4];9.9286→H[10,4]
182: 11.7143→H[11,4];13.5273→H[12,4];15.3519→H[13,4];17.2037→H[14,4]
183: 20.9074→H[15,4];24.6111→H[16,4];28.3148→H[17,4];32→H[18,4]
184: 35.5263→H[19,4];39→H[20,4]
185: gto +15
186: .9259→H[1,4];1.9293→H[2,4];2.4343→H[3,4];2.9394→H[4,4];4→H[5,4]
187: 5.0556→H[6,4]
188: for N=7 to 14;5.0555+2.7778(N-6)→H[N,4];next N
189: for N=15 to 20;32.8333+5.5556(N-15)→H[N,4];next N
190: gto +10
191: .561→H[1,4];1.1249→H[2,4];1.4127→H[3,4];1.7004→H[4,4];2.276→H[5,4]
192: 2.8515→H[6,4];4.3275→H[7,4];5.8256→H[8,4];7.3445→H[9,4];8.8881→H[10,4]
193: 10.4638→H[11,4];12.1044→H[12,4];13.7611→H[13,4];15.4627→H[14,4]
194: for N=15 to 20;18.8804+3.4176(N-15)→H[N,4];next N
195: gto +5
196: .9112→H[1,4];1.9002→H[2,4];2.3941→H[3,4];2.8912→H[4,4];3.9281→H[5,4]
197: 4.9698→H[6,4]
198: for N=7 to 14;7.702+2.73375(N-7)→H[N,4];next N
199: for N=15 to 20;32.3056+5.4675(N-15)→H[N,4];next N
  401
```

```
200: 20→Z
201: for N=0 to 29;0→H[N,5]→H[N,6];next N
202: A[1]→H[-1,5]→F[-1,1];A[2]→H[-1,6]→F[-1,2]
203: A[3]→H[-2,5];A[4]→H[-2,6]
204: A[1]→H[0,6];0→H[8,3]→H[9,3];A[1]→H[10,3]
205: 1→H[1,3]→P
206: "ENTER OBSERVATION TIME:"→C$;gsb "DI"
207: L→H[11,3];if L<.01 or L>150;gto -1
208: "ENTER ELEV ANGLE AT TIME:"→C$[1,25];str(H[11,3])→C$[26]
209: gsb "DI"
210: L→H[12,3];if L<0 or L>90;gto -2
211: spc 1;prt "TOP OF ZONE DATA","  TIME:",H[11,3]
212: fxd 2;prt " EL ANGLE",H[12,3]
213: "ENTER AZ ANGLE AT TIME:  "→C$[1,25];str(H[11,3])→C$[26];gsb "DI"
214: L→H[13,3];if L<0 or L>360;gto -1
215: fxd 2;prt " AZ ANGLE",H[13,3];spc 2
216: "ARE PRINTED DATA OK ?"→C$[1,32];gsb "Y/N"
217: if A=-2tn^10;prt "---REPEATING----";gto -11
218: H[11,3]→T
219: if A[5]=30;gsb "H30"
220: if A[5]=31;gsb "HE30"
221: if A[5]=100;gsb "H100"
222: if A[5]=101;gsb "HE100"
223: gto +55
224: "H100":
225: if T<=1;356.6T→H
226: if T<=3 and T>1;356.6+347.5(T-1)→H
227: if T<=5 and T>3;1051.6+338.3(T-3)→H
228: if T<=8 and T>5;1728.2+329.2(T-5)→H
229: if T<=10 and T>8;2715.8+320(T-8)→H
230: if T<=11 and T>10;3355.8+310.9(T-10)→H
231: if T<=14 and T>11;3666.7+301.8(T-11)→H
232: if T<=36 and T>14;4572+292.6(T-14)→H
233: if T>36;11009.4+301.8(T-36)→H
234: if T>=38 and T<=42;H-9.1→H
235: if T>=42 and T<=45;H-18.3→H
236: if T>45 and T<=48;H-27.4→H
237: if T>48 and T<=52;H-36.6→H
238: if T>52 and T<=55;H-45.7→H
239: if T>55 and T<=58;H-54.9→H
240: if T>58;H-64→H
241: ret
242: "H30":
243: if T<=1;219.5T→H
244: if T<=3 and T>1;219.5+201.2(T-1)→H
245: if T<=5 and T>3;621+192(T-3)→H
246: if T>5;1005+182.9(T-5)→H
247: ret
248: "HE100":
249: if T<=1;350T→H
*22567
```

```
250: if T<=2 and T>1;350+320(T-1)→H
251: if T<=3 and T>2;670+310(T-2)→H
252: if T<=4 and T>3;980+305(T-3)→H
253: if T<=5 and T>4;1285+300(T-4)→H
254: if T<=6 and T>5;1585+295(T-5)→H
255: if T<=7 and T>6;1880+290(T-6)→H
256: if T<=9 and T>7;2170+285(T-7)→H
257: if T<=12 and T>9;2740+280(T-9)→H
258: if T<=15 and T>12;3580+275(T-12)→H
259: if T<=30 and T>15;4405+270(T-15)→H
260: if T<=32 and T>30;8455+275(T-30)→H
261: if T<=34 and T>32;9005+280(T-32)→H
262: if T<=37 and T>34;9565+285(T-34)→H
263: if T<=38 and T>37;10420+290(T-37)→H
264: if T<=45 and T>38;10710+295(T-38)→H
265: if T<=50 and T>45;12775+300(T-45)→H
266: if T<=51 and T>50;14275+295(T-50)→H
267: if T<=52 and T>51;14570+290(T-51)→H
268: if T<=53 and T>52;14860+285(T-52)→H
269: if T<=60 and T>53;15145+280(T-53)→H
270: if T>60;17105+282.5(T-60)→H
271: ret
272: "HE30":
273: if T<=1;216T→H
 74: if T<=3 and T>1;216+198(T-1)→H
275: if T<=5 and T>3;612+189(T-3)→H
276: if T>5;990+180(T-5)→H
277: ret
278: if H>H[P,1];gto +2
279: H[11,3]→H[8,3];H[12,3]→H[9,3];H[13,3]→H[10,3];cfg 0;gto -73
280: (H[8,3]-H[P,4])/(H[8,3]-H[11,3])→L
281: if H[10,3]-H[13,3]>180;H[13,3]+360→H[13,3]
282: if H[13,3]-H[10,3]>180;H[10,3]+360→H[10,3]
283: H[10,3]-L(H[10,3]-H[13,3])→H[P,5]
284: if H[P,5]>360;H[P,5]-360→H[P,5]
285: H[9,3]-L(H[9,3]-H[12,3])→H[P,6]
286: if flg0;gto +2
287: gto +14
288: H[-1,5]→A;H[-1,6]→R;gsb "P→R"
289: X→r1;Y→r2
290: H/tan(H[12,3])→R
291: Rsin(H[13,3])→Y
292: Rcos(H[13,3])→X
293: gsb "R→P"
294: X-r1→X;Y-r2→Y
295: .03238X/(H[11,3]-H[8,3])→U;.03238Y/(H[11,3]-H[8,3])→V
296: UL→U;VL→V
297: U/.03238→X;V/.03238→Y
298: X+r1→X;Y+r2→Y;gsb "R→P"
299: if H>=H[P,1];A→H[P,5];atn(H[P,1]/R)→H[P,6]
  611
```

```
300: if H[P,5]>360;H[P,5]-360→H[P,5]
301: P+1→P→H[1,3];if not flg0;gto +2
302: if H<H[P,1];cfg 0
303: if P<Z;gto -25
304: if P<29;for I=P to 29;0→H[I,4];next I
305: prt "******************"
306: if flg3;gto +29
307: prt "ZONE INFORMATION"
308: spc 3;fxd 0;prt "SURFACE"
309: prt " HOR DST",H[-1,6]
310: fxd 3;prt " AZ OFF",H[-1,5]
311: fxd 0;prt " ALTmsl",val(P$[19,21])10
312: fxd 0;prt " WDIR,deg",H[-2,5]
313: fxd 0;prt " WSPEED,K",H[-2,6]
314: H[-1,5]→A;H[-1,6]→R;gsb "P→R"
315: for N=1 to 29;if H[N,4]=0 or (H[N,5] and H[N,6])=0;gto +19
316: N-1→J
317: 6367650→S
318: √((S+H[N,1])^2-S^2cos(H[N,6])^2)-Ssin(H[N,6])→r3
319: (Scos(H[N,6])/(S+H[N,1]))r3→R
320: H[N,5]→A;gsb "P→R"
321: r1→r4;X→r1;r4-r1→X;r2→r5;Y→r2;r5-r2→Y;gsb "R→P"
322: (R/(H[N,4]-H[J,4])).03238→S;17.778A→A
323: if N=1 and H[N,4]<1;H[N,4]S→S
324: H[J,1]+(H[N,1]-H[J,1])/2+val(P$[19,21])10→H
325: spc 1;fxd 0;prt "ALTmsl",H
326: fxd 0;prt "WDIR,mils",A
327: fxd 0;prt "WSPEED,k",S
328: spc 1;fxd 0;prt "ZONE",N
329: prt "ALTgeom",H[N,1]
330: fxd 3;prt " TIME",H[N,4]
331: fxd 3;prt " ELEV",H[N,6]
332: fxd 3;prt " AZ",H[N,5]
333: next N
334: prt "******************";spc 5
335: for N=0 to 29;0→H[N,2];next N
336: ent "ENTER REGION NUMBER",R
337: fxd 0;prt "REGION",R;spc 3
338: if R[2]>0 and R[2]<4;7→T
339: if R[2]>3 and R[2]<7;1→T
340: if R[2]>6 and R[2]<10;3→T
341: if R[2]>9;5→T
342: if O<=800 or O>1800;T+1→T
343: trk 0;ldf 2,Z$
344: if 10val(P$[19,21])<200;1→D
345: if 10val(P$[19,21])>=200;5→D;6→R
346: for N=0 to 25
347: if val(Z$[D,8N+1,8N+2])=R and val(Z$[D,8N+3,8N+4])=T;gto +2
348: next N
349: val(Z$[D,8N+7,8N+8])*10→E
*1522
```

```
350: val(Z$[D+1,8N+1,8N+2])/10+P→B
351: val(Z$[D+1,8N+3,8N+6])/10→E
352: if A[8]<B;B→A[8]
353: if A[8]>E+.5;E+.5→A[8]
354: int((A[8]-B+.5)/.5)→C
355: if C<=10;C→F;trk val(Z$[D+1,8N+7,8N+8]);ldf val(Z$[D+2,8N+1,8N+2]),Z$
356: if C>=11 and C<=20;C-10→F
357: if C>=11 and C<=20;val(Z$[D+2,8N+3,8N+4])→O;val(Z$[D+2,8N+5,8N+6])→W
358: if C>=11 and C<=20;trk O;ldf W,Z$
359: if C>=21 and C<=30;C-20→F
360: if C>=21 and C<=30;val(Z$[D+2,8N+7,8N+8])→O;val(Z$[D+3,8N+1,8N+2])→W
361: if C>=21 and C<=30;trk O;ldf W,Z$
362: if C>=31;C-30→F;trk val(Z$[D+3,8N+3,8N+4]);ldf val(Z$[D+3,8N+5,8N+6]),Z$
363: for N=0 to 25;prt Z$[F,8N+1,8N+8]
364: next N
365: spc 5
366: for N=0 to 60;for M=1 to 6;0→F[N,M];next M;next N
367: 100→F[1,6];350→F[2,6];750→F[3,6];1250→F[4,6];1750→F[5,6]
368: 2250→F[6,6];2750→F[7,6];3250→F[8,6];3750→F[9,6];4250→F[10,6]
369: 4750→F[11,6];5500→F[12,6];6500→F[13,6];7500→F[14,6];8500→F[15,6]
370: 9500→F[16,6];10500→F[17,6];11500→F[18,6];12500→F[19,6];13500→F[20,6]
371: 14500→F[21,6];15500→F[22,6];16500→F[23,6];17500→F[24,6];18500→F[25,6]
372: 19500→F[26,6]
373: A[7]→F[0,1];A[11]→F[0,2];A[9]→F[0,3];A[6]→F[0,4]
 74: I→M
375: for N=1 to 26
376: val(Z$[F,8(N-1)+1,8(N-1)+4])/10→F[N,3]
377: val(Z$[F,8(N-1)+5,8(N-1)+8])→F[N,4]
378: F[N,3]-273.16→F[N,1]
379: if H[N,1]<=F[M,6] or H[N,4]=0;gto +4
380: (F[M,6]-H[N-1,1])/(H[N,1]-H[N-1,1])→A
381: H[N-1,4]+A(H[N,4]-H[N-1,4])→F[M,5]
382: M+1→M
383: next N
384: A[6]→H[-1,2];A[7]+273.16→H[-1,4]
385: 0→I;log(A[6])→H[0,2];1→H[0,3]
386: for P=1 to 29
387: P-1→J
388: if P=2 or P=4 or P=5;gto +8
389: if P=1;0→J
390: if P=3;1→J
391: if P=6;3→J
392: 1+I→I
393: if F[I,4]=0;gto +4
394: 2log(F[I,4])-H[J,2]→H[P,2]
395: (H[J,2]-H[P,2])67.442(F[I,3]+.04I)+H[J,1]→H[P,1]
396: next P
397: "N"→P$[100,100]
398: for N=1 to 26;if F[N,5]=0;N→Y[1];N-1→Y[2];gto +2
399: next N
 2154
```

```
400: 1+H[-2,3]→H[-2,3]
401: trk 1;rcf 1,P$,Q$,O$,L$,N$,G$,D$
402: trk 1;rcf 2,R[*],Y[*]
403: trk 1;rcf 3,H[*],D[*]
404: trk 1;ldp 4
405: end
*26027
```

c. Output

```
OUTPUT #1 ON TRACK 1 FILE 4
             TRACK 1 FILE 4


0: dsp "TURN-ON REMEX";gsb "BBB"
1: dim A$[100],B$[5],R[10],C[0:16,1:14]
2: dim P$[102],Q$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
3: dim F[-1:60,1:6],Y[2]
4: dim H[-2:29,1:6],D[0:29],B[15]
5: dim H$[2,32];time 500
6: trk 1;ldf 1,P$,Q$,O$,L$,N$,G$,D$
7: spc 5;prt "PIBAL FLIGHT:",D$[1,16]
8: sfg 4;if not ios2;cfg 4
9: if not ios2;gto +6
10: ent "NEED TTY-76 TELETAPE?",A
11: if A#-1tn^10 and A#-2tn^10;gto -1
12: if A=-1tn^10;"Y"→P$[101,101]
13: if A=-2tn^10;"N"→P$[101,101]
14: trk 1;rcf 28,P$,Q$,O$,L$,N$,G$,D$
15: spc 5
16: dsp "PIBAL FLIGHT MESSAGE OUTPUT";gsb "BBB"
17: if P$[101,101]="Y";sfg 7;cfg 8
18: if P$[101,101]="N";cfg 7;sfg 8
19: trk 1;ldf 2,F[*],Y[*]
20: trk 1;ldf 3,H[*],D[*]
21: " 5 9 #,. )4&80:;3 $? 6!/-2  71( "→H$[1]
22: " T O HNM LRGIPCVEZDBSYFXAWJ UQK "→H$[2]
23: char(0)→H$[1,1,1];char(10)→H$[1,9,9];char(7)→H$[1,21,21]
24: char(13)→H$[1,3,3]
25: gsb "INTHT"
26: sfg 3;dsp "SOUND RANGING";gsb "BBB"
27: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
28: gto +69
29: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
30: "INTHT":for N=1 to Y[1];if F[N,3]#0;gto +7
31: for K=N to Y[1];if F[N,3]=0;next K
32: (F[K,5]-F[N,5])/(F[K,5]-F[N-1,5])→C
33: F[K,1]-C(F[K,1]-F[N-1,1])→F[N,1]
34: F[K,2]-C(F[K,2]-F[N-1,2])→F[N,2]
35: F[K,3]-C(F[K,3]-F[N-1,3])→F[N,3]
36: F[K,6]-C(F[K,6]-F[N-1,6])→F[N,6]
37: next N
38: ret
39: "FOOT":fmt 2,"g",/
40: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2;wtc 2,0
41: fmt b,z
42: if flg4 and flg7;par 0;wtc 2,2;wrt 2,3,2,8;wtc 2,0
43: if flg4;wtc 2,2;for N=1 to 30;wtb 2,0;next N;wtc 2,0
44: ret
45: "A→B":fmt b,z;par 0
46: for G=1 to B
47: for K=1 to 2;pos(H$[K],A$[G,G])→F;if F;sfg K;gto +2
48: next K
49: if F=1 or F=3 or F=5 or F=9;cfg 1,2,5,6;gto +5
*1266
```

```
50: if flg1 and flg5;cfg 1;gto +4
51: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
52: if flg2 and flg6;cfg 2;gto +2
53: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
54: wtc 2,2;wrt 2,F-1;wtc 2,0
55: next G
56: wtc 2,2;wrt 2,2,8;wtc 2,0
57: ret
58: "R→P":√(XX+YY)→R[1]
59: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
60: A→R[2]
61: if A<0;360+A→R[2]
62: ret
63: "HEADER":cfg 1,2,5,6;par 2
64: if flg4 and flg8;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
65: if flg4 and flg7;12→B;gsb "A→B"
66: dsp A$[1,12];prt A$[1,12]
67: P$[85,99]→A$[1,J]
68: if flg9;cfg 9;gto +5
69: fmt 3,c12
70: if flg4 and flg8;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
71: if flg4 and flg7;12→B;gsb "A→B"
72: gto +3
73: if flg4 and flg8;fmt 3,c15;wtc 2,2;wrt 2.3,A$[1,15];wtc 2,0
74: if flg4 and flg7;15→B;gsb "A→B"
75: dsp A$[1,J];prt A$[1,J]
76: ret
77: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000
78: ret
79: "P→R":P[1]*cos(P[2])→X;P[1]*sin(P[2])→Y
80: ret
81: "STRING":fxd 0;str(M)→B$[1,5]
82: if M<1000;B$[2,4]→B$[3,5];"0"→B$[2,2]
83: if M<100;B$[2,4]→B$[3,5];"0"→B$[2,2]
84: if M<10;B$[2,4]→B$[3,5];"0"→B$[2,2]
85: if M<1;B$[2,4]→B$[3,5];"0"→B$[2,2]
86: ret
87: "ARRAY":1→M;gsb "STRING"
88: B$[4,5]→A$[1,2];A→M;gsb "STRING"
89: B$[3,5]→A$[3,5];S→M;gsb "STRING"
90: B$[3,5]→A$[6,8];T→M;gsb "STRING"
91: B$[2,5]→A$[9,12];O→M;gsb "STRING"
92: B$[2,5]→A$[13,16]
93: if flg4 and flg8;par 2;fmt 4,c16;wtc 2,2;wrt 2.4,A$[1,16];wtc 2,0
94: if flg4 and flg7;16→B;gsb "A→P"
95: dsp A$[1,16];prt A$[1,16]
96: ret
97: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→P"
98: X→R[4];Y→P[6]
99: prnd(H[-2,5]*16/9,0)→A→B[10]
*30887
```

```
100: prnd(H[-2,6],0)→S→B[9]
101: prnd(H[-1,4]10,0)→T
102: .0000001→H[0,4]
103: 0→I
104: if H[5,4]=0;spc 5;prt "SOUND RANGING   NOT READY";gto +49
105: for P=1 to 5;if H[P,4]=0;gto +14
106: P-1→J
107: if P=3;next P
108: if P=4;2→J
109: 1+I→I
110: 6367650→S
111: √((S+D[P])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
112: (Scos(H[P,6])/(S+D[P]))R[8]→R[1]
113: H[P,5]→R[2];gsb "P→R"
114: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "P→P"
115: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S→B[I]
116: if P=1 and H[P,4]<1;H[P,4]S→S→B[I]
117: prnd(R[2]*16/9,0)→A→B[I+4]
118: next P
119: if B[2]<B[1] and (B[2]>=B[9]+2 or B[2]<=B[9]-2);B[2]→S;B[6]→A;gto +16
120: if B[2]<B[1] and (B[2]<B[9]+2 and B[2]>B[9]-2);B[1]→S;B[5]→A;gto +15
121: if B[2]>2*B[1];.4→r1;0→r2;.3→r3;.15→r4;.15→r5;gto +2
122: .2→r1;.5→r2;.15→r3;.075→r4;.075→r5
123: 0→X→Y→C→D
124: r1B[9]→R[1];B[10]*9/16→R[2];gsb "P→R"
125: X→C;Y→D
126: r2B[1]→R[1];B[5]*9/16→R[2];gsb "P→R"
127: X+C→C;Y+D→D
128: r3B[2]→R[1];B[6]*9/16→R[2];gsb "P→R"
129: X+C→C;Y+D→D
130: r4B[3]→R[1];B[7]*9/16→R[2];gsb "P→R"
131: X+C→C;Y+D→D
132: r5B[4]→R[1];B[8]*9/16→R[2];gsb "P→R"
133: X+C→X;Y+D→Y;gsb "P→P"
134: F[1]→S;R[2]*16/9→A
135: prnd(S,0)→S;prnd(A,0)→A
136: if A=0;64→A
137: if S=0;0→A
138: for L=1 to 5;if not (F[L,6]>200 and F[L-1,6]<200);next L
139: (200-F[L-1,6])/(F[L,6]-F[L-1,6])→O
140: O(F[L,3]-F[L-1,3])+F[L-1,3]→H[16,3]
141: O(F[L,1]-F[L-1,1])+F[L-1,1]+273.16+3H[16,3]→H[16,3]
142: H[16,3]/4-273.16→H[16,3]→T
143: prnd(T*10,0)→T
144: abs(T)→M;" "→A$[1,1];if T<0;"-"→A$[1,1]
145: gsb "STRING"
146: B$[3,5]→A$[2,4];A→M;gsb "STRING"
147: B$[3,5]→A$[5,7];S→M;gsb "STRING"
148: B$[4,5]→A$[8,9]
149: if not flg3;gto +5
*31774
```

```
150: A$[1,9]→P$[91,99];P$[73,84]→A$[1,12];15→J
151: sfg 9;gsb "HEADER"
152: cfg 9;gsb "FOOT"
153: spc 5
154: sfg 3;dsp "COMPUTER MET";gsb "BBB"
155: H[-1,5]→P[2];H[-1,6]→R[1];gsb "P→R"
156: X→R[4];Y→R[6]
157: str(prnd(H[-1,2],0))→A$;A$[2,4]→P$[22,24]
158: if H[-1,2]>=1000;A$[3,5]→P$[22,24]
159: P$[1,12]→A$[1,12];P$[13,24]→P$[85,96];12→J
160: if not flg3;gto +35
161: gsb "HEADER"
162: 0→I
163: prnd(H[-2,5]*16/9,0)→A;if A=0;64→A
164: prnd(H[-2,6],0)→S;if S=0;0→A
165: prnd(H[-1,4]10,0)→T
166: prnd(H[-1,2],0)→C
167: gsb "ARRAY"
168: .0000001→H[0,4]
169: for P=1 to 29;if H[P,4]=0;gto +24
170: if P=1;0→J
171: if P=2;next P
172: if P=3;1→J
173: if P=4 or P=5;next P
174: if P=6;P-3→J
175: if P>6;P-1→J
176: 1+I→I
177: 6367650→S
178: √((S+D[P])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
179: (Scos(H[P,6])/(S+D[P]))R[8]→R[1]
180: H[P,5]→R[2];gsb "P→R"
181: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "P→P"
182: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S
183: if P=1 and H[P,4]<1;H[P,4]S→S
184: H[0,3]H[P,1]6371299/(6371299+H[P,1])→R[1]
185: H[0,3]H[J,1]6371299/(6371299+H[J,1])→R[2]
186: prnd(((R[1]-R[2])/67.442(H[J,2]-H[P,2]))10,0)→T
187: tn^((H[P,2]+H[J,2])/2)→C;prnd(C,0)→C
188: prnd(P[2]*16/9,0)→A
189: if A=0;64→A
190: if S=0;0→A
191: gsb "ARRAY"
192: next P
193: gsb "FOOT"
194: spc 5
195: sfg 3;dsp "FALLOUT MET";gsb "BBB"
196: str(prnd(1000H[-1,2]/1013.25,0))→A$;A$[2,4]→P$[22,24]
197: if H[-1,2]>=1013.25;A$[3,5]→P$[22,24]
198: gto +11
199: "ARRAY-12":I→M;gsb "STRING"
*11279
```

```
200: B$[4,5]→A$[1,2];A→M;gsb "STRING"
201: B$[4,5]→A$[3,4];S→M;gsb "STRING"
202: B$[4,5]→A$[5,6];T→M;gsb "STRING"
203: B$[3,5]→A$[7,9];Q→M;gsb "STRING"
204: B$[3,5]→A$[10,12]
205: if flg4 and flg8;par 2;fmt 3,cl2;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
206: if flg4 and flg7;12→B;gsb "A→P"
207: dsp A$[1,12];prt A$[1,12]
208: ret
209: H[-1,5]→R[2];H[-1,6]→P[1];gsb "P→R"
210: X→R[4];Y→R[6]
211: 1→I
212: H[-1,2]→C[0,4]
213: for P=1 to 27;if H[P,4]=0;gto +24
214: P-2→J
215: if P=1;0→J
216: if P=2;next P
217: if P=3;1→J
218: if P=4 or P=5;next P
219: if P=6;P-3→J
220: if P=7 or P=8 or P=15;P-1→J
221: if P=9 or P=11 or P=13 or P=16 or P=18;next P
222: if P=20 or P=22 or P=24 or P=26;next P
223: 6371299→S
224: √((S+D[P])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
225: (Scos(H[P,6])/(S+D[P]))R[8]→R[1]
226: H[P,5]→R[2];gsb "P→R"
227: R[4]→R[5];X→R[4];P[5]-R[4]→X;R[6]→R[7];Y→R[6];P[7]-R[6]→Y;gsb "R→P"
228: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S→C[I,2]
229: if P=1 and H[P,4]<1;H[P,4]S→S→C[I,2]
230: H[0,3]H[P,1]6371299/(6371299+H[P,1])→B[1]
231: H[0,3]H[J,1]6371299/(6371299+H[J,1])→B[2]
232: (B[1]-B[2])/67.442(H[J,2]-H[P,2])→C[I,3];prnd(C[I,3],0)→T
233: tn^((H[P,2]+H[J,2])/2)→Q→C[I,4];prnd(Q,0)→Q
234: R[2]→C[I,5]
235: if I<=14;I+1→I→C[I,1]
236: next P
237: for Z=1 to 16;if C[Z,4]#0;next Z
238: Z-1→Z
239: H[-1,4]→C[0,3]
240: for N=0 to Z;348.38395C[N,4]/C[N,3]→C[N,9];next N
241: C[0,9]/1225→C[0,10];C[1,9]/1213.3→C[1,10];C[2,9]/1184.4→C[2,10]
242: C[3,9]/1139.2→C[3,10];C[4,9]/1084.6→C[4,10];C[5,9]/1032→C[5,10]
243: C[6,9]/957→C[6,10];C[7,9]/863.4→C[7,10];C[8,9]/777→C[8,10]
244: C[9,9]/697.4→C[9,10];C[10,9]/590→C[10,10];C[11,9]/467→C[11,10]
245: C[12,9]/364.8→C[12,10];C[13,9]/266.6→C[13,10];C[14,9]/194.8→C[14,10]
246: C[15,9]/142.3→C[15,10]
247: 100C[0,10]→C[0,11]
248: H[-1,4]→C[0,3]
249: C[0,3]/288.2→C[0,6]
*8800
```

```
250: C[1,3]/287.5→C[1,6];C[2,3]/285.9→C[2,6];C[3,3]/285.3→C[3,6]
251: C[4,3]/280→C[4,6];C[5,3]/276.8→C[5,6];C[6,3]/271.9→C[6,6]
252: C[7,3]/265.5→C[7,6];C[8,3]/259→C[8,6];C[9,3]/252.5→C[9,6]
253: C[10,3]/242.7→C[10,6];C[11,3]/229.8→C[11,6];C[12,3]/216.8→C[12,6]
254: C[13,3]/216.7→C[13,6];C[14,3]/216.7→C[14,6];C[15,3]/216.7→C[15,6]
255: gto +9
256: "ARRAY-FO":I→M;gsb "STRING"
257: B$[4,5]→A$[1,2];A→M;gsb "STRING"
258: B$[3,5]→A$[3,5];S→M;gsb "STRING"
259: B$[3,5]→A$[6,8]
260: if flg4 and flg8;par 2;fmt 1,c8;wtc 2,2;wrt 2.1,A$[1,8];wtc 2,0
261: if flg4 and flg7;8→P;gsb "A→B"
262: dsp A$[1,8];prt A$[1,8]
263: ret
264: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→R"
265: X→P[4];Y→R[6]
266: prnd(H[-2,5]*16/9,0)→A;if A=0;64→A
267: prnd(H[-2,6],0)→S;if S=0;0→A
268: P$[49,60]→A$[1,12];P$[61,72]→P$[85,96];12→J
269: if not flg3;gto +29
270: gsb "HEADER"
271: 0→I
272: gsb "ARRAY-FO"
273: 1→I
274: for P=8 to 34;if H[P,4]=0;gto +22
275: if P=8;0→J
276: if P=9 or P=10 or P=11;next P
277: if P=12;8→J
278: if P=13 or P=14;next P
279: if P=15;12→J
280: if P=16 or P=18 or P=20 or P=22 or P=24 or P=26 or P=28;next P
281: if P=17 or P=19 or P=21 or P=23 or P=25 or P=27 or P=29;P-2→J
282: if P>29;P-1→J
283: 6371299→S
284: √((S+D[P])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
285: (Scos(H[P,6])/(S+D[P]))R[8]→P[1]
286: H[P,5]→R[2];gsb "P→R"
287: R[4]→P[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "P→R"
288: prnd((P[1]/(H[P,4]-H[J,4])).03238,0)→S
289: if P=1 and H[P,4]<1;H[P,4]S→S
290: prnd(P[2]*16/9,0)→A
291: if A=0;64→A
292: if S=0;0→A
293: gsb "ARRAY-FO"
294: 1+I→I
295: next P
296: gsb "FOOT"
297: spc 5
298: sfg 3;dso "BALLISTIC TYPE 3";gsb "BPB"
299: gto +11
*26704
```

```
300: "T3":
301: E+1→E;for N=1 to Z;rNC[E,6]+C[N,7]→C[N,7];next N
302: ret
303: "D3":
304: E+1→E;for N=1 to Z;rNC[E,10]+C[N,11]→C[N,11];next N
305: ret
306: "WW":
307: C[I,2]→R[1];C[I,5]→R[2];1+I→I;gsb "P→R"
308: for N=1 to Z;XrN+C[N,13]→C[N,13];YrN+C[N,14]→C[N,14];next N
309: ret
310: 100C[0,6]→C[0,7]
311: 100→r1;27→r2;13→r3;8→r4;5→r5;4→r6;2→r7;1→r8→r9→r10→r11→r12→r13→r14→r15
312: gsb "T3"
313: 0→r1;73→r2;20→r3;12→r4;10→r5;4→r6→r7;3→r8;gsb "T3"
314: 0→r2;67→r3;25→r4;20→r5;9→r6;7→r7;5→r8;2→r9→r10→r11→r12→r13→r14→r15
315: gsb "T3"
316: 0→r3;55→r4;21→r5;11→r6;9→r7;4→r8;3→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
317: 0→r4;44→r5;13→r6;11→r7;10→r8;gsb "T3"
318: 0→r5;59→r6;26→r7;19→r8;9→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
319: 0→r6;41→r7;23→r8;13→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
320: 0→r7;35→r8;24→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
321: 0→r8;44→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
322: 100→r1;43→r2;22→r3;15→r4;11→r5;8→r6;6→r7;5→r8;4→r9;3→r10;1→r11
323: 2→r12→r13→r14→r15;0→F;gsb "D3"
324: 0→r1;57→r2;31→r3;21→r4;17→r5;11→r6;8→r7;6→r8;6→r9;4→r10;3→r11→r12→r14
325: 4→r15;gsb "D3"
326: 0→r2;47→r3;32→r4;25→r5;17→r6;14→r7;11→r8;9→r9;7→r10;5→r11→r12→r14→r15
327: 4→r13;gsb "D3"
328: 0→r3;32→r4;22→r5;17→r6;13→r7;5→r13;gsb "D3"
329: 0→r4;25→r5;15→r6;12→r7;10→r8;8→r9;6→r11;gsb "D3"
330: 0→r5;32→r6;22→r7;19→r8;17→r9;13→r10;12→r11;11→r12→r13;10→r14→r15;gsb "D
331: 0→r6;25→r7;17→r8;15→r9;12→r10;11→r11;10→r12;9→r13→r14→r15;gsb "D3"
332: 0→r7;21→r8;14→r9;11→r10;9→r11→r12;8→r14→r15;gsb "D3"
333: 0→r8;18→r9;11→r10;9→r11;8→r12→r13;7→r14→r15;gsb "D3"
334: 0→r9;25→r10;16→r11;14→r12→r13;13→r14;12→r15;gsb "D3"
335: 0→r10;23→r11;12→r12;10→r13;11→r14;9→r15;gsb "D3"
336: 0→r11;16→r12;9→r13;8→r14→r15;gsb "D3"
337: 0→r12;12→r13;6→r14;5→r15;gsb "D3"
338: 0→r13;8→r14;5→r15;gsb "D3"
339: 0→r14;6→r15;gsb "D3"
340: 1→I
341: 100→r1;20→r2;9→r3;6→r4;4→r5;3→r6;2→r7→r8→r9;1→r10;0→r11→r12→r15;gsb "WW
342: 0→r1;80→r2;19→r3;12→r4;6→r5;5→r6;3→r7;2→r10;1→r12→r13→r14→r15;gsb "WW"
343: 0→r2;72→r3;26→r4;15→r5;8→r6;7→r7;6→r8;5→r9;1→r11;gsb "WW"
344: 0→r3;56→r4;20→r5;9→r6;4→r10→r11;2→r12;gsb "WW"
345: 0→r4;53→r5;12→r6;8→r7;3→r10→r11;4→r12;3→r13;2→r14→r15;gsb "WW"
346: 0→r5;63→r6;20→r7;14→r8;12→r9;7→r10→r12→r13→r14→r15;8→r11;gsb "WW"
347: 0→r6;53→r7;19→r8;13→r9;8→r10→r11;7→r12;gsb "WW"
348: 0→r7;45→r8;20→r9;9→r10→r11;gsb "WW"
349: 0→r8;36→r9;8→r12;gsb "WW"
*339
```

```
350: 0→r9;55→r10;20→r11;17→r12;15→r13;13→r14;12→r15;gsb "WW"
351: 0→r10;38→r11;16→r12;14→r13;gsb "WW"
352: 0→r11;30→r12;13→r13;11→r15;gsb "WW"
353: 0→r12;24→r13;10→r14→r15;gsb "WW"
354: 0→r13;18→r14;8→r15;gsb "WW"
355: 0→r14;14→r15;gsb "WW"
356: for N=1 to Z;C[N,13]→X;C[N,14]→Y;gsb "R→P"
357: R[1]→C[N,13];R[2]→C[N,14];next N
358: H[-2,5]→C[0,14];H[-2,6]100→C[0,13]
359: if not flg3;gto +14
360: P$[25,36]→A$[1,12];P$[13,24]→P$[85,96];12→J;gsb "HEADER"
361: for N=0 to Z;N→I
362: prnd(C[N,11]10,0)→Q
363: prnd(C[N,7]10,0)→T
364: prnd(C[N,13]/100,0)→S
365: prnd(C[N,14]*16/90,0)→A
366: if A=0;64→A
367: if S=0;0→A
368: if S>99;S-100→S;I+80→I
369: gsb "ARRAY-12"
370: next N
371: gsb "FOOT"
372: spc 5
373: trk 1;ldf 5
374: end
*26786
```

```
OUTPUT #2 IN TRACK 1 FILE 5
 TRACK 1 FILE 5


0: sfg 3;dsp "BALLISTIC TYPE 2";gsb "BBB"
1: gto +79
2: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
3: "FOOT":fmt 2,"9",/
4: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2;wtc 2,0
5: fmt b,z
6: if flg4 and flg7;par 0;wtc 2,2;wrt 2,3,2,8;wtc 2,0
7: if flg4;wtc 2,2;for N=1 to 30;wtb 2,0;next N;wtc 2,0
8: ret
9: "A→P":fmt b,z;par 0
10: for G=1 to B
11: for K=1 to 2;pos(H$[K],A$[G,G])→F;if F;sfg K;gto +2
12: next K
13: if F=1 or F=3 or F=5 or F=9;cfg 1,2,5,6;gto +5
14: if flg1 and flg5;cfg 1;gto +4
15: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
16: if flg2 and flg6;cfg 2;gto +2
17: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
18: wtc 2,2;wrt 2,F-1;wtc 2,0
19: next G
20: wtc 2,2;wrt 2,2,8;wtc 2,0
21: ret
22: "R→P":√(XX+YY)→R[1]
23: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
24: A→R[2]
25: if A<0;360+A→R[2]
26: ret
27: "HEADER":cfg 1,2,5,6;par 2
28: if flg4 and flg8;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
29: if flg4 and flg7;12→B;gsb "A→P"
30: dsp A$[1,12];prt A$[1,12]
31: P$[85,99]→A$[1,J]
32: if flg9;gto +4
33: if flg4 and flg8;wtc 2,2;wrt 2.3,A$[1,J];wtc 2,0
34: if flg4 and flg7;12→B;gsb "A→P"
35: gto +3
36: if flg4 and flg8;fmt 3,c15;wtc 2,2;wrt 2.3,A$[1,J];wtc 2,0
37: if flg4 and flg7;15→B;gsb "A→P"
38: dsp A$[1,J];prt A$[1,J]
39: ret
40: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000
41: ret
42: "P→R":R[1]*cos(R[2])→X;R[1]*sin(R[2])→Y
43: ret
44: "STRING":fxd 0;str(M)→B$[1,5]
45: if M<1000;B$[2,4]→B$[3,5];"0"→B$[2,2]
46: if M<100;B$[2,4]→B$[3,5];"0"→B$[2,2]
47: if M<10;B$[2,4]→B$[3,5];"0"→B$[2,2]
48: if M<1;B$[2,4]→B$[3,5];"0"→B$[2,2]
49: ret
*23874
```

```
50: "ARRAY":I+[illegible] "STRING"
51: E$[4,5]+A$[1,2];A+M;gsb "STRING"
52: B$[3,5]+A$[3,5];S+M;gsb "STRING"
53: B$[3,5]+A$[6,8];T+M;gsb "STRING"
54: B$[2,5]+A$[9,12];Q+M;gsb "STRING"
55: B$[2,5]+A$[13,16]
56: if flg4 and flg8;par 2;fmt 4,cl6;wtc 2,2;wrt 2.4,A$[1,16];wtc 2,0
57: if flg4 and flg7;16+B;gsb "A+B"
58: dsp A$[1,16];prt A$[1,16]
59: ret
60: "ARRAY-12":I+M;gsb "STRING"
61: E$[4,5]+A$[1,2];A+M;gsb "STRING"
62: B$[4,5]+A$[3,4];S+M;gsb "STRING"
63: B$[4,5]+A$[5,6];T+M;gsb "STRING"
64: B$[3,5]+A$[7,9];Q+M;gsb "STRING"
65: B$[3,5]+A$[10,12]
66: if flg4 and flg8;par 2;fmt 3,cl2;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
67: if flg4 and flg7;12+B;gsb "A+B"
68: dsp A$[1,12];prt A$[1,12]
69: ret
70: "T2":
71: E+1+F;for N=1 to Z;rNC[E,6]+C[N,8]+C[N,8];next N
72: ret
73: "D2":
74: E+1+E;for N=1 to Z;rNC[E,10]+C[N,12]+C[N,12];next N
75: ret
76: "WW":
77: C[1,2]+R[1];C[1,5]+R[2];1+I+I;gsb "P+E"
78: for N=1 to Z;XrN+C[N,13]+C[N,13];YrN+C[N,14]+C[N,14];next N
79: ret
80: 100C[0,6]+C[0,8]
81: 100+r1;63+r2;37+r3;25+r4;20+r5;13+r6;10+r7;9+r8;7+r9;5+r10+r11+r13+r14
82: 4+r12;5+r15;0+F;gsb "T2"
83: 0+r1;37+r2;37+r3;30+r4;24+r5;19+r6;14+r7;10+r8;9+r9;8+r10
84: 6+r11+r12+r13+r14+r15;gsb "T2"
85: 0+r2,26+r3;35+r4;30+r5;24+r6;20+r7;17+r8;14+r9;12+r10;10+r11+r12+r13+r14
86: 10+r15;gsb "T2"
87: 0+r3;10+r4;18+r5+r6;16+r7;15+r8;13+r9;10+r10;8+r12
88: 9+r11+r13+r14+r15;gsb "T2"
89: 0+r4+8+r5;14+r6+r7;13+r8;12+r9;10+r10;8+r11+r13+r14+r15;gsb "T2"
90: 0+r5,12+r6;19+r7;20+r8;19+r9;17+r10;15+r11;14+r12;16+r13+r14+r15;gsb "T2"
91: 0+r6;7+r7;12+r8;15+r9;14+r10;13+r11+r12;12+r13+r14+r15;gsb "T2"
92: 0+r7;4+r8;8+r9;10+r10;12+r11;11+r12;13+r13+r14+r15;gsb "T2"
93: 0+r8;3+r9;8+r10;10+r11+r12;11+r13+r14+r15;gsb "T2"
94: 0+r9;6+r10;12+r11;16+r12;10+r13+r14+r15;gsb "T2"
95: 100C[0,10]+C[0,12]
96: 100+r1;63+r2;37+r3;25+r4;20+r5;13+r6;10+r7;9+r8;7+r9;5+r10;4+r11+r12
97: 3+r13+r14;2+r15;0+E;gsb "D2"
98: 0+r1;37+r2;37+r3;30+r4;24+r5;19+r6;14+r7;10+r8;9+r9;8+r10;6+r11+r12
99: 5+r13+r14+r15;gsb "D2"
*32459
```

```
100: 0→r2;26→r3;35→r4;30→r5;24→r6;20→r7;17→r8;14→r9;12→r10;10→r11;9→r12
101: 8→r13;6→r14→r15;gsb "D2"
102: 0→r3;10→r4;18→r5→r6;16→r7;15→r8;13→r9;10→r10;8→r11→r12
103: 7→r14→r15;gsb "D2"
104: 0→r4;8→r5;14→r6→r7;13→r8;12→r9;6→r13;5→r15;gsb "D2"
105: 0→r5;12→r6;19→r7;20→r8;19→r9;17→r10;15→r11;13→r12;12→r13;11→r14→r15
106: gsb "D2"
107: 0→r6;7→r7;12→r8;15→r9;14→r10;13→r11;12→r12;11→r13;10→r14→r15;gsb "D2"
108: 0→r7;4→r8;8→r9;10→r10→r11→r12→r13;9→r14;8→r15;gsb "D2"
109: 0→r8;3→r9;8→r10;10→r11;8→r12→r13→r14→r15;gsb "D2"
110: 0→r9;6→r10;12→r11;13→r12→r13→r14→r15;gsb "D2"
111: 0→r10;4→r11;7→r12;9→r13;10→r14→r15;gsb "D2"
112: 0→r11;2→r12;5→r13;6→r14;7→r15;gsb "D2"
113: 0→r12;2→r13;4→r14;5→r15;gsb "D2"
114: 0→r13;1→r14;3→r15;gsb "D2"
115: for N=1 to Z;0→C[N,13]→C[N,14];next N
116: 1→I
117: 100→r1;50→r2;29→r3;18→r4;13→r5;8→r6;7→r7;4→r8→r9;3→r10→r12;1→r15
118: 2→r11→r13→r14;gsb "WW"
119: 0→r1;50→r2;33→r3;23→r4;18→r5;12→r6;8→r7→r8;6→r9;3→r15
120: 4→r10→r11→r12→r13→r14;gsb "WW"
121: 0→r2;38→r3;39→r4;31→r5;22→r6;16→r7;13→r8;11→r9;8→r10;6→r11;7→r12
122: 5→r13→r14→r15;gsb "WW"
123: 0→r3;20→r4;27→r5;20→r6;15→r7;12→r8;10→r9;7→r11;6→r13→r14;4→r15;gsb "WW"
124: 0→r4;11→r5;19→r6;16→r7;13→r8;6→r11;4→r14;5→r15;gsb "WW"
125: 0→r5;19→r6;27→r7;24→r8;21→r9;16→r10;13→r11;12→r12;11→r13;9→r14→r15
126: gsb "WW"
127: 0→r6;11→r7;18→r8;20→r9;15→r10;11→r12;10→r13;gsb "WW"
128: 0→r7;8→r8;12→r9;14→r10;12→r11;10→r12;9→r13;8→r15;gsb "WW"
129: 0→r8;6→r9;13→r10;11→r11;8→r12→r13→r14;7→r15;gsb "WW"
130: 0→r9;11→r10;18→r11;15→r12;14→r13;13→r14;12→r15;gsb "WW"
131: 0→r10;8→r11;10→r12;11→r13→r14;10→r15;gsb "WW"
132: 0→r11;6→r12;9→r13→r14→r15;gsb "WW"
133: 0→r12;5→r13;6→r14;8→r15;gsb "WW"
134: 0→r13;5→r14;6→r15;gsb "WW"
135: 0→r14;4→r15;gsb "WW"
136: for N=1 to Z;C[N,13]→X;C[N,14]→Y;gsb "P→P"
137: P[1]→C[N,13];P[2]→C[N,14];next N
138: H[-2,5]→C[0,14];H[-2,6]100→C[0,13]
139: if not flg3;gto +14
140: P$[37,48]→A$[1,12];P$[13,24]→P$[85,96];12→J;gsb "HEADER"
141: for N=0 to Z;N→I
142: prnd(C[N,12]10,0)→C
143: prnd(C[N,8]10,0)→T
144: prnd(C[N,13]/100,0)→S
145: prnd(C[N,14]*16/90,0)→A
146: if A=0;64→A
147: if S=0;0→A
148: if S>99;S-100→S;I+80→I
149: gsb "ARRAY-12"
*10299
```

```
150: next
151: asb "FOOD"
152: spc 5
153: trk 0:idn 0,0,5
154: end
*9296
```

d. Departure Table Input Program

```
[illegible]
[illegible]
[illegible]


0: dim A$[10,200],X[4],Y[4]
1: dim D$[9],C$[7],D$[30],F$[20]
2: trf 0;lck 1
3: "LIST TABLES ?"→D$;gsb "Y/N"
4: if A=-1tr^10;gsb "LISTTABLES"
5: ent "STACK NUMBER",X
6: if X#0 and X#1;gto -1
7: ent "ITEM NUMBER",Y
8: if Y<1 or Y>66;gto -1
9: ent "NUMBER OF LINES",I
10: trl X;lcf Y,A$
11: "CORRECT TABLES ?"→D$;gsb "Y/N"
12: if A=-1tr^10;gsb "CORRECT"
13: gsb "INPUT"
14: stp
15: "Y/N":
16: csp D$;ent "",A
17: if A#-1tr^10 and A#-2tr^10;gto -1
18: ""→D$
19: ret
20: "INPUT":
21: ent "COLUMN NUMBER",Z
22: if Z<1 or Z>10;gto -1
23: if Z=-2tr^10;ret
24: for N=0 to I-1
25: ent "ITEMTABLE",A$[Z,8N+1,8N+8]
26: prt N+1,A$[Z,8N+1,8N+8]
27: next N
28: trl X;rcf Y,A$
29: gto -8
30: ret
31: "CORRECT":
32: ent "COLUMN NUMBER",Z
33: if Z<1 or Z>10;gto -1
34: if Z=-2tr^10;ret
35: ent "ITEM NUMBER",N
36: N-1→N
37: ent "NEWTABLE",A$[Z,8N+1,8N+8]
38: prt N+1,A$[Z,8N+1,8N+8]
39: trl X;rcf Y,A$
40: gto -8
41: ret
42: "LISTTABLES":
43: trl 0;lcf 2,A$
44: ent "PROCESS",F
45: ent "STARTING ITEM",J
46: if I<200;1→C
47: if F>=200;5→C
48: ent "NUMBER OF LINES",J
49: if J<1 or J>8;gto -1
*12755
```

```
50: for N=1 to 2?
51: val(A$[C,8N+1,8N+2])→X
52: val(A$[C,8N+3,8N+4])→Y
53: if X=0 or Y=0;next I
54: val(A$[C,8N+5,8N+6])→[illegible]
55: val(A$[C,8N+7,8N+8])→[illegible];1(I)→I
56: val(A$[C+1,8N+1,8N+2])/1(C+1)→[illegible]
57: val(A$[C+1,8N+3,8N+6])/1(C+1
58: val(A$[C+1,8N+7,8N+8])→X[1]
59: val(A$[C+2,8N+1,8N+2])→Y[1]
60: val(A$[C+2,8N+3,8N+4])→X[2]
61: val(A$[C+2,8N+5,8N+6])→Y[2]
62: val(A$[C+2,8N+7,8N+8])→X[3]
63: val(A$[C+3,8N+1,8N+2])→Y[3]
64: val(A$[C+3,8N+3,8N+4])→X[4]
65: val(A$[C+3,8N+5,8N+6])→Y[4]
66: if Q=1;"SPRING DAYTIME  "→F$
67: if Q=2;"SPRING NIGHTTIME"→F$
68: if Q=3;"SUMMER DAYTIME  "→F$
69: if Q=4;"SUMMER NIGHTTIME"→F$
70: if Q=5;"FALL   DAYTIME  "→F$
71: if Q=6;"FALL   NIGHTTIME"→F$
72: if Q=7;"WINTER DAYTIME  "→F$
73: if Q=8;"WINTER NIGHTTIME"→F$
74: for M=1 to 4
75: if X[M]=0 and Y[M]=0;gto [illegible]-22
76: [illegible] X[M],Y[M],A$
77: for [illegible] 1
78: fmt 1,[illegible]x,c20,f2.0,c20
79: wrt [illegible].1,"[illegible] REGION ",F,F$
80: fmt 1,[illegible]x,c40
81: wrt [illegible],"[illegible] SPEED FOR COMPUTER TIME-PERIOD"
82: wrt 6;wrt 6
83: wrt [illegible].1,"[illegible] SURFACE HUMIDITY-EFFECT"
84: fmt 1,z,c10
85: wrt 6.1,"          "
86: for N=1 to 5
87: fmt 1,z,f7.1," to ",f5.1
88: wrt 6.1,.5(N-1)+[illegible]5(N-1)+2.5?,.5(N-1)+[illegible](N-1)+2.5?+.4
89: next N
90: wrt 6
91: fmt 1,z,c10
92: wrt 6.1," ZONE NO. "
93: for I=1 to [illegible]
94: "     ------"→[illegible][1,9]
95: "     -----"→C$[1,7]
96: fmt 1,z,c9,c7
97: wrt 6.1,[illegible],C$
98: next I
99: wrt 6
*1[illegible]
```

```
100: for I=1 to 2[illegible]
101: fmt 1,[illegible],f10.[illegible]
102: wrt 6.1,[illegible]
103: for J=1 to [illegible]
104: fmt 1,[illegible],f1[illegible].[illegible],f7.[illegible]
105: wrt 6.1,Va[illegible]([illegible]+5[illegible],[illegible]+1,[illegible]+[illegible]))/1[illegible],Va[illegible]([illegible]+5[illegible],[illegible],[illegible]+[illegible]))
106: next J
107: wrt 6
108: next I
109: stp
110: next [illegible]
111: next [illegible]
112: gto -[illegible]
113: ret
114: end
[illegible]
```

6 REFERENCES

FM 6-15, Artillery Meteorology, Headquarters, Department of the Army, AUG 78.

FM 6-16, Tables for Artillery Meteorology, Headquarters, Department of the Army, MAY 79.

HP 09825-90000. Operating and Programming Manual for the Hewlett Packard 9825A Calculator. 76

DEP TM 11-6660-205-10, Operator's Manual, Meteorological Data Processing Group OL-192(*), US Army Electronics Command, Fort Monmouth, NJ.

DELCS-TR-79-1, Documentation of Software in the OL-192 Meteorological Data Reduction Program, Feb 79, US Army Electronics Command, Ft Monmouth, NJ.


FIL